

\$14.95

# GAMES APPLES PLAY

050N<sup>OPT</sup> 133  
PRICE 14.95  
A 41<sup>TM</sup> 0454



By Mark James Capella and Michael D. Weinstock

Learn programming the fun enjoyable way...by gaming! Included is a vast selection of classic games for your Apple written in Applesoft BASIC. Why make programming hard work?







# **GAMES APPLES PLAY**







# **GAMES APPLES PLAY**

by

**Mark James Capella**

and

**Michael D. Weinstock**

Commentaries on Games Listings by

**Scott L. Singer**

Cover Art and Illustrations by

**Art Huff**



8943 Fullbright Avenue  
Chatsworth, California 91311  
(213) 709-1202



Reston Publishing Company, Inc.  
*A Prentice-Hall Company*  
Reston, Virginia  
ISBN 0-8359-2417-3

This book is published and copyrighted by DATAMOST. All rights are reserved by DATAMOST. Copying, duplicating, selling or otherwise distributing this product is hereby expressly forbidden except by prior written consent of DATAMOST.

The word APPLE and the Apple logo are registered trademarks of APPLE COMPUTER, INC.

APPLE COMPUTER, INC. was not in any way involved in the writing or other preparation of this book, nor were the facts presented here reviewed for accuracy by that company. Use of the term APPLE should not be construed to represent any endorsement, official or otherwise, by APPLE COMPUTER, INC.



## ACKNOWLEDGMENTS

To my wife, Genie, for all those times she turned her head to see me sneaking up the stairs to the computer room.

Mark James Capella

To Diana and Aaron. Diana for her patience, understanding and the cold dinners she ate while waiting for me to finish just a little bit more. Aaron for the thrill of understanding what it means having a four year old around when things just aren't working correctly.

To Brian Davis for his creative artistic talents that greatly assisted our effort.

To Scott Singer for his enlightening commentary on the programs.

To Dave Gordon because he is Dave Gordon, a trusted friend.

Michael D. Weinstock





## Table of Contents

Introduction -- Using Games to Learn BASIC .....	9
SCI-FI .....	17
Apple Learner .....	33
Biorhythm .....	37
Connect Five .....	43
Digits .....	49
Grue Stew .....	55
IRSMan .....	63
Kingdom .....	69
Magic Squares .....	75
Numbers Away .....	81
Reverser .....	89
Transition .....	93
Word Scramble .....	97
MUBBLE CHASE .....	105
Air Attack .....	127
Artist Board .....	133
Barrel of Fun .....	137
Block 'Em .....	145
Brain Teaser .....	151
Brick Wall .....	157
Craps .....	163
Dragon's Lair .....	169
Hang Man .....	177
Itche .....	183
Knock Out .....	189
Leaky Faucet .....	195
Match the Key .....	199
Miniature Golf .....	205
Moving Target .....	217
Point Attack .....	221
Robot Chase .....	225
Safe Cracker .....	231
Saucer Duels .....	237
Schmoo .....	243
Stardodger .....	249
Stranded .....	255
Target .....	259
Twinky .....	263

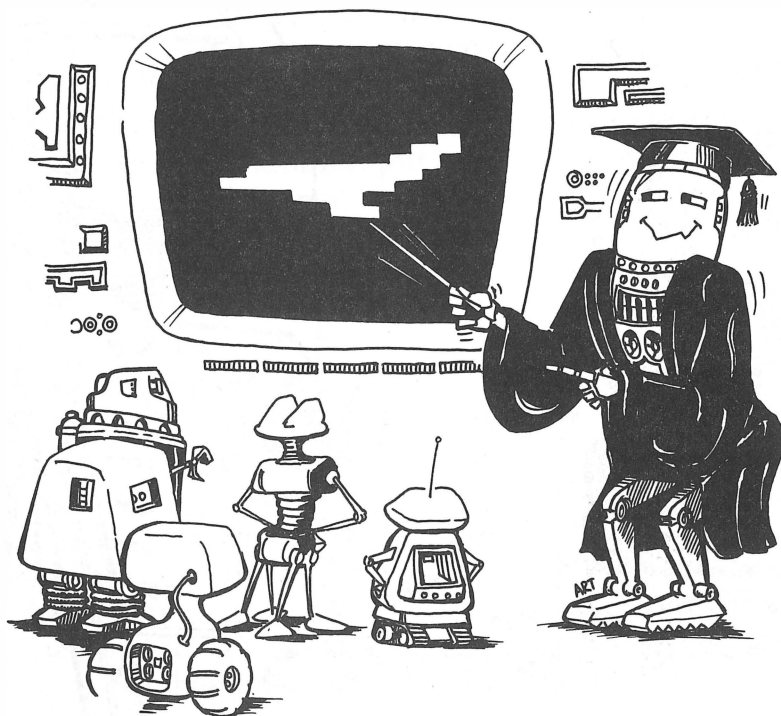




# INTRODUCTION

## Using games to learn BASIC

Commercially written games for the Apple computer are now being written in machine language with elaborate copy protection schemes. Techniques to increase speed and foil pirates also have the effect of making games both hard to understand and hard to learn from. Such was not always the case. In the good old days of 16K Apples and cassette tapes, the games were given away at user groups. Most of the games were in Integer BASIC. The listings were published in magazines and newsletters, and seldom exceeded two pages. Novice programmers didn't care that you could drink an entire soda pop before the ship got across the screen or that you could watch the bombs falling in slow motion, blip——blip——blip. Most of the people using those games wanted to learn about programming color graphics, and games were a good way to learn. Author Mike Weinstock has compiled a selection of the classic games such as AIR ATTACK and SAUCER DUELS that have been the precursor of many of the faster arcade games. Many types of games are not dependent on speed, such as adventure games like DRAGON'S LAIR, word games like HANG MAN and board games such as CONNECT-FIVE. These are all included along with many clever new games written by the author.





The games are written in Applesoft BASIC in a structured format. Structuring makes the programs more understandable and allows you to easily lift or adapt any routines that you like for inclusion in your own programs. The workings of each game are explained in a way that allows you to modify and customize the games to your heart's content. Two of the games, Mubble Chase and Sci Fi, are examined line by line. Sci-Fi introduces the principles of text formatting, word games, data statements, and input routines. Mubble Chase will help you understand how grids are constructed and graphic figures are moved and detected.

Most of the programs in this book are structured in the same way, with a GOSUB, an empty target line and a RETURN line in exactly the same place. This allows you to start writing the 'action' parts of the program first and fill in the title screen, instructions, and ending routines later. This is a complete outline program that will RUN but doesn't do anything:

```
10 REM STRUCTURED PROGRAM
20 GOSUB 1000: REM INSTRUCTIONS
30 GOSUB 2000: REM SETUP
40 GOSUB 3000: REM PLAY
50 GOSUB 4000: REM END
60 END
1000:
1990 RETURN
2000:
2990 RETURN
3000:
3990 RETURN
4000:
4990 RETURN
```

You can use this program as an outline for your own original programs. It helps you to stay organized and reminds you of what parts need finishing. Give this program a snappy title such as EMPTY and save it on a diskette. Load it into memory when you feel creative and start writing within the framework.

Techniques are given here that allow you to dissect basic programs and see what makes them tick. The games themselves are hours of fun. You will learn a lot about programming and de-bugging by typing in the games from the book and making them run correctly, or you may order the games diskette from DataMost, and use dissecting techniques described here to investigate their inner workings. Either way, you will not find a more enjoyable way to learn BASIC!

Do not be afraid of your computer! Remember, you are smart and the computer is dumb. The reason that programs have bugs in them is that the computer has to be told every little thing. It can't figure out a misspelled word the way you can. Fortunately Applesoft was written by some very clever humans who tried to program the computer to forgive as many errors as possible and to give very clear error messages for mistakes it does find. Relax and have fun. NOTHING you type in from the keyboard can hurt the workings of the computer, you will just get what is called GIGO—garbage in—garbage out.

Whatever is typed into the computer's memory or loaded in from a tape or disk drive exists in a temporary state. If you turn off the computer or the dog trips over the power cord the program is gone—vaporized. If the program came from a tape or diskette it is still there and can be re-loaded as if nothing happened. We emphasize this because we want you to play around with the programs, change the lines around, put in silly statements and eventually get it so messed up that you will want to throw it away and load in the original. This is the best way to learn. Try these techniques for investigating the inner workings of programs:

### **Change lines**

As an example, if you see `COLOR = X`, change it to `color = 8` (red). Run the program and you will see that some figure that is supposed to change color now just stays red; often figures are drawn in one color, redrawn in black, and drawn again in color one space over. This makes the figure appear to move. If the figure becomes a red streak on the screen you have discovered the line that moves the figure. You will also know when the line is used in the program and what is affected by the variation you introduced. Change variables (A B X Y etc.) to arbitrary numbers. Do things get stuck? Do you get overflow or 'illegal

quantity' errors? Variables are often the hardest things to understand. After a line containing a variable 'A' (or any variable) you can add a print statement :PRINT A . A number representing the value of 'A' will flash on the screen when the line gets executed and this number may give a clue to the function of variable 'A' (we don't guarantee it).

### **Use TRACE**

Just type this magic word before you run the program and the line numbers that are being executed are displayed at the bottom of the screen. Watch for repeated series; this is the program loop where the action occurs. TRACE stays on until you enter NOTRACE or RESET.

### **Use STOP**

Put in STOP in a separate line, giving it a number between two numbered lines in the program. The program will stop at that point and wait for you to type CONT(inue). By using STOP you can tell what parts of the program execute before STOP is reached.



## Detour

If you want to know what the program will do without line 100, put a GOTO in front of it sending the program around line 100 without ever executing it.

```
99 GOTO 110 (add this line)
100 E=MC*MC (mystery line)
110 PRINT "Hello"
```

Now you can see what the program does without line 100. Does it still run? What goes wrong? Restore the program by typing 99 with nothing after it. This deletes both the line and the line number that you added.

## Delete lines

A quicker technique than the detour. Just type the line number and RETURN to eliminate the line from the program. Use this method when the line numbers are too close together for detouring or have lots of GOTO and RETURN. Just reload the program from diskette to undo all your surgical mishaps.

## Add a 'beep'

If you want to know when a line is executed add a line `PRINT CHR$(7)` immediately after it. Be sure the line above doesn't send the program somewhere else with a `GOTO` or `RETURN`. If it does, rewrite the line with `PRINT CHR$(7):` at the beginning of the mystery line. The computer will obligingly beep when the line is executed.

### **Isolate parts of programs**

Take away all but a few lines of the program by use of `DEL(ete)` 100,1200 or whatever line numbers are appropriate. Get the portion that is left to work by adding a few lines of your own. Mix parts of programs together using the Renumber program that is part of the DOS master diskette that comes with the Apple. Renumber has a `MERGE` utility that lets you put one program on hold, load another program, and merge the two together. You may get crazy results, but a lot can be learned in the process. You can name and save a part of a program even if it doesn't run. Leave yourself little notes (`REMs`) to remind you where the pieces came from.

## Add paddle or keyboard input

If you want to see the action of variable 'X' you can add a line

```
NNN (line number) X = PDL(0)
```

You have to experiment around to find the right place for this line in the program loop. You will be able to control some action of the program that was previously automatic. Whatever value the program assigned to X up this point will be replaced by the X you insert. The paddle returns a value from 0 to 255. If this number is too large and makes the program crash, the value of paddle (0) can be divided to yield just the right range of values:

```
X = INT (PDL (0) / 6.5)
```

This command will give you INTegers (whole numbers with no decimals) from 0 to 39 to use with the lo-res graphics screen that is 40 characters wide. Of course additional variables can be controlled with paddle (1) and keyboard inputs until the entire program loop is under your control.

## Save the programs you have changed around

When you get something that works, save it as an intermediate version even if you want to continue making changes. Often we make some useful changes and then mess the program up with later additions. You can delete extra versions later on if you run short of disk space. The only way to lose a program on diskette is to save another program with the same name on the same diskette. SO USE A DIFFERENT NAME!

Don't forget that you must turn on the computer with a DOS diskette in drive number one in order to be able to save programs to diskette. This is called BOOTING DOS. You can check to see if DOS is there by typing CATALOG. Always keep a few initialized empty diskettes handy, because the diskette must be initialized before it can be used. Here is a handy little HELLO program that will display the catalog of the diskette every time it is booted:

Turn on the Apple with the DOS Master diskette in drive one. When the ] prompt appears type NEW

```
10 TEXT:HOME
20 D$= CHR$(4):REM THIS IS CONTROL D
30 PRINT "JOHN DOE'S GAME DISKETTE"
40 PRINT "TODAY'S DATE"
50 PRINT D$;"CATALOG"
```



Run the program to test it out, and then place a new diskette in drive one. Be sure the diskette is empty because initializing a diskette erases everything that was on it. Add this line to the HELLO program:

```
60 NEW
```

Don't run the program after you have added this line, because line 60 erases the HELLO program from memory and leaves the computer 'empty' for your program to be entered.

Type INIT HELLO <RETURN>

Apple will create an initialized diskette that will display its catalog when booted and then clear the decks for your programs. This diskette will load DOS and allow you to save programs. You will not have to use the DOS MASTER diskette first.

Before the individual programs are explained, copy and run the following program.

```
10 GR
20 FOR I = 0 TO 15
30 COLOR= I
40 VLIN 0,34 AT 2*I+3
50 NEXT
60 PRINT " * * * * * * * * * 1 1 1 1 1 1"
70 PRINT " 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5"
```

You will note that above the zero (0) there is, apparently, a blank space. Actually, the color zero is black (the background color), so what you see (or don't see) is a black line drawn on a black background.

Fill up your program diskette and have fun.

A note about BUGS! It is not inconceivable that a few mistakes have crept into the following listings. Trust your intuition and tinker with the program even if you have to change what is in the book. Please drop us a card if you find a real boo-boo.

This program is not a game, rather, it is an entertaining collection of short stories which you make as personal as you want. You are given the chance to enter your own data. As with MUBBLE CHASE, this program will be described line by line.

10 REM stands for REMark. Any comments, numbers, symbols, expletives, or anything else may follow REM. In this case, the remark is used to highlight the program name. The computer, in effect, ignores the material which follows REM. In this case, the

REMark \*\*\*\*\* is made.

11-16 completes the title with REM statements including two blank lines for readability.

20 The entire program is controlled by lines 20 through 50. There are actually two parts to line 20. First, GOSUB 1000 instructs the computer to go to line 1000, and to continue from there until the command RETURN is encountered. The second instruction, REM, is allowed on the same line for only one reason. The colon (:) allows the programmer to clump many, not necessarily related, instructions onto the same command line. This is usually done to show that certain pieces of a program are so closely related, that to isolate them on separate lines would belie their common purpose or association. The REMark: INSTRUCTIONS is the programmer's way of telling the reader that the subroutine beginning at line 1000 (GOSUB 1000) is where you can find the INSTRUCTIONS. In this case, the remark serves to explain the purpose of the subroutine rather than serving the equally important, but more eccentric, role of lines 10 through 16.

30 When the subroutine, begun by line 20, is completed by the RETURN statement, the program, having completed line 20, drops down to the next

sequential instruction, which in this case is line 30. Like line 20, this line first initiates a subroutine (GOSUB 2000), and then informs the reader as to the main emphasis of the subroutine, which is the program SET-UP.

40 This line is identical, in function, to line 30.

50 END returns control to the machine. Control WAS with the program.

1000 This line is the 'target' of the GOSUB in line 20.

1001 Here, the REMark: \*\*\* INSTS is again made to inform the reader of the function of the subroutine.

1010 This line contains three, distinct instructions. TEXT instructs the computer to change to (if not already in) the text mode. In this mode, all forty lines are reserved for text. The use of color is reserved for the graphics mode. Next, the instruction NORMAL is encountered. This command instructs the computer to display the text using white letters on a black background. HOME clears the screen of all text. Instructions such as this are called 'housekeeping', and should be included in all your programs.

1020 VTAB is the programmer's way of telling the computer to Vertically TAB down three spaces. The reason that VTAB instructs the computer to tab DOWN, is that in terms of X and Y, 0,0 is in the UPPER left-hand corner. After the computer Vertically TABs down three (3) lines from the top of the screen, it is told to Horizontally TAB thirteen (13) spaces to the right. The next instruction, PRINT, tells the computer to prepare to output whatever follows. If the material is written between a set of quotes, then whatever is enclosed by the quotes is printed verbatim. If there are no quotes, then the numeric value of the variable will be printed. To illustrate, if a program reads: 10 X = 5  
20 PRINT "X+3" the output will read 'X+3'. On the other hand, if a program reads: 10 X = 5 20 PRINT X+3 the output will be '8'. Line 1020 causes the message \*\*\* SCI-FI \*\*\* to be printed. It will begin at a position three lines from the top (VTAB 3) and thirteen columns from the left margin (HTAB 13).

1030 After Vertically Tabbing down seven (7) lines, the message that appears between the quotes will be printed. This time, the message will begin at the left-hand margin.

1040 The empty PRINT statement serves a very useful purpose. What this statement does is to PRINT a blank line. Notice that some of the words of text in the listings are split in the middle and continue on the next line, but when printed on the screen by the program the text is neatly formatted. In the listing

there is no space between HOPESOF but it prints correctly when run. HOPE is at the end of the line, so OF would be indented one space when printed by the program if a space occurred in the listing. Good looking screen formats are a matter of trial and error. If you try to edit print statements using the arrow key you might have noticed that seven blank spaces are inserted in the text whenever the cursor wraps around a line. There is a cure for this problem that allows quick editing of basic listings: type POKE 33,33 <RETURN> before typing LIST. The text will not be indented and can be easily copied over and parts changed. RESET gets things back to normal.

1050 Notice that the instructions asks you to type RETURN. Nothing gets entered in ANS\$ and in fact it becomes an 'empty' string. The function of ANS\$ in this program is just to hold up the works until you have read the screen and want to continue. Many programs ask for your name in a similar situation and then use the input in an appropriate response.

Try this:

```
1050 VTAB 23: INPUT "HI, WHAT'S YOUR NAME";NA$
1060 PRINT"WELL HELLO ";NA$;"", LETS PLAY SCI-FI"
1070 FOR I= 1 TO 2000:NEXT I
```

Since the program would dash off after it received NA\$ we add a delay loop in 1070 to wait just long enough for us to read line 1060. You can use any letters for variables as long as they are not basic commands (reserved words). For clarity variables should suggest what they perform. Programmers generally use ANS for answer and NA for names but are not required to. The ON—GOTO command can provide excellent flexibility in your programs. In Sci-Fi a random number sends the program off to different sections, but this command also works well for branching from a menu:

```
10 PRINT "PRESS 1 FOR SALAD, 2 FOR ENTREE, OR 3 FOR
DESSERT"
20 GET X
30 ON X GOTO 100,200,300
```

Starting at line 100 you would put the salad choices, etc. If you succeed in actually teaching the computer to make a salad let me know.

1990 RETURN ends the subroutine initiated by line 20 and begun at line 1000. At this point, program flow is RETURNed to line 20, and then line 30.

2000 As stated before, a blank colon (:) is a legitimate means of writing a virtually blank line within the program itself.



2001 On this line, the main function of the subroutine is detailed by a REM statement.

2002 This command line serves to separate line 2001 from the body of the text.

2005 This line appears to be a nebulous conglomeration of variables. Not so! This line serves a very specific purpose. DEF stands for DEFine. The next question is, what is to be defined? The answer is, a FuNction (FN). The FuNction being DEFined is R(X). 'R' is the given name of the function. 'X' is a variable name which is equal to the FIRST INT ( RND (1) \* X) + 1. Each time the FuNction 'R (any variable or digit)' is used, the variable inside the parentheses assumes the value of 'INT ( RND (1) \* (the new variable or digit)) + 1. Following will be a list which, hopefully, will help you to understand the 'DEF FN' statement.

<u>FUNCTION</u>	<u>VALUE OF VARIABLE OR DIGIT</u>	<u>VALUE OF FUNCTION</u>
A(X)=4*X+5	'X', IF NOT STATED, =0	A(X)=5 (4*0+5)
A(13)	13=13	A(13)=57 (4*13+5)
A(Y)	LET US SAY THAT Y=6	A(Y)=29 (4*6+5)
BS(X)=X*X-22	X=0	BS(X)=-22 (0*0-22)
BS(17)	17 = 17	BS(17)=267 (17*17-22)
BS(FN A(Y))	FN A(Y)=29	BS(FN A(Y))=819
A(FN BS(17))	FN BS(17)=267	A(FN BS(17))=1073 (267*4+5)

Here is a program to further illuminate the function of line 2005.

<u>THE PROGRAM</u>	<u>THE OUTPUT</u>
10 A = 17 : C = 2.65	—
20 DEF FN PRY(C) = -A * C	—
30 PRINT C; : PRINT FN PRY(C)	2.65 -45.05
40 PRINT A; : PRINT FN PRY(A)	17 -289
50 PRINT FN PRY(FN PRY(C))	765.85

Line 20 identifies the variable as being 'C'. Therefore, whenever the FuNction 'PRY' is executed, the variable (or digit) within the parentheses is substituted for C. In line 2005, 'X' is the variable. If the number five (5) is substituted for X, then the result is, R(5) = INT ( RND (1) \* 5 + 1. The reason the DEF FN instruction is used in this program is so that whenever a random number between 1 and any other number is needed, all the programmer need write is R(any other number), and the random result will be generated.

2010 This line sets aside sixteen memory locations for SO\$. The instruction, DIM, instructs the computer to DIMension memory so as to allow for sixteen separate values of SO\$. Also, the value of SO is set to zero. The number in parentheses is the number of the array variable. Apple starts counting from (0).

2011 This line sets up a one-dimensional table in memory. This table can accomodate up to sixteen separate values of PL\$. Also, PL is set to 0.

2012 and 2013 are both duplicates of lines 2010 and 2011.

2015 The READ statement is an interesting animal. What it does is to find the first available DATA statement and read from it. In this program, the first DATA statement is at line 2020. What happens is, the first piece of data before a comma (Alexander Haig) is read into (stored at) SO\$. Then a test is done to see if SO\$ is equal to END. If so, the program falls through to line 2016. If not, SO is incremented, SO\$(0) is assigned the value of the contents of SO\$ (Alexander Haig), and then the process is repeated (GOTO 2015). The program will next READ the second piece of DATA (Ronald Reagan) and store it in SO\$. The test will again prove negative, SO will be incremented, RONALD REAGAN will be stored at SO\$(1), and the process will be repeated. After the seven pieces of DATA on line 2020 are read, the DATA statement on 2021 is read next. This time the test on line 2015 (SO\$ < > "END") will prove to be affirmative, so the program will fall through to line 2016. When you understand that SO, SO\$ and SO\$(0) are all different variables you get a gold star.

2016 Once a DATA statement has been read, it is no longer "available". Therefore, the first available data is on line 2025. It seems likely that PL stands for PLace. Actually, all of the variables are representative of their meaning.

2017 and 2018 Both of these lines are identical in function to line 2015.

2020 If the data in a DATA statement is to be read into a character-string location (a variable ended with a dollar sign), then it must be enclosed in quotes (" "). Each item is kept separate from other items by using a comma.

2021 through 2036 These are all DATA statements. All of the data could have been combined into one long DATA statement. The reason for dividing the lines was to add clarity.

2100 (six instructions) HOME clears the screen of text. The VTAB and HTAB instructions pinpoint where the beginning print location will be. The message, `*** SCI=FI ***`, is outputted. The computer is next instructed to TAB down to line 22. At this point a message is printed.

2110 The messages printed at line 2100 still appear on the screen. Previously, the printing had been done on line 22 (VTAB 22). The first instruction brings the computer back up to line 7 (VTAB 7). The message between the quotes is printed, starting at line seven at the left-hand margin. The empty PRINT statement follows the above message with a blank line.

2114 This line sets the counter (CO) to zero.

2115 Line 2110 revealed the nature of the input, now you have a place to put those names. Just input whatever you'd like. When you've entered five pieces of data, or when you enter nothing, the program will fall through to line 2117. The data that you input is stored in SO\$ (temporarily). Then the input is tested to see if "nothing" (RETURN) was entered. If so, the program falls through. If not, the line continues; CO is incremented; SO\$ is moved to a permanent location (SO\$(SO)); the counter is incremented and tested; if the counter is not yet five, then the line is begun again.

2117 This line clears the screen (HOME), and then prints the message beginning at line three (VTAB 3) column thirteen (HTAB 13). Then the computer is instructed to tab down to line seven (VTAB 7).

2120-2145 continues the INPUT sequences for Places that will be Attacked, Names of Monsters, etc.

2150 RETURN ends the subroutine initiated by line 30 and started at line 2000. Program control is returned to line 30.

2900 This line sets the value of PT to one (1) each time the subroutine (lines 2900-2990) is begun.

2905 MID\$ states that the computer will look at the MIDDLE of a given word. Which word? All of the remaining pertinent information is in the parentheses. The word or words that are to be checked by MID\$ are contained in WRD\$. Beginning at the left-hand side of WRD\$, the computer will begin the scrutiny at that character plus PT. If WRD\$ contained the phrase "I Love You", and PT equals six, then MID\$ (WRD\$,PT) would instruct the computer to begin looking at 'I Love You' six characters over from the left most character (the 'e' in 'Love' is the SIXth character). If not specified, the computer will begin the

search at the designated location, and continue through to the end of the word/s. If you want a certain number of characters looked at, just specify that number after you tell the computer where to start. The following program should help.

<u>PROGRAM</u>	<u>OUTPUT</u>
10 A\$ = "SEND MONEY"	--
20 PRINT MID\$ (A\$,4,5)	D MON
30 PRINT MID\$ (A\$,1,6)	SEND M

In line 20, the computer is told to search the MIDDle of A\$, to begin 4 characters from the left and to PRINT the next 5 characters. Likewise, line 30 tells the computer to PRINT the contents of A\$ beginning with character 1, and to continue for a total of 6 characters. You will admit that at the end of most words you will find a blank space. The test (IF MID\$ (WRD\$,PT,1) = " ") checks to see if the character at position PT is a blank space ( ) and it checks to see if its line position is greater than 30. PEEK (36) surveys the screen and checks to see if the cursor is beyond column 30 (there are 40 columns to a row (0-39)). If the cursor is beyond column 30, and MID\$ (WRD\$,PT,1) = " ", then the next word to be written is in danger of overflowing the right-hand margin. To prevent this from happening, an affirmative test result will force the computer to skip to the next line before continuing to print.



2910 There are two steps to this line. First, PT is incremented ( $PT = PT + 1$ ). Second, the size of PT is compared to the character LENGTH of WRD\$. For example, if WRD\$ contained the phrase EARTH WAS ATTACKED, then the LENGTH of WRD\$ would equal the number of characters in that phrase (18). To prevent lines 2905 and 2910 from looping indefinitely, there must be some

contingency factor, some restraint which stops the loop. This is the function of the second half of line 2910. When PT is larger than the length of WRD\$, the program will fall through to line 2990.

2990 This line completes the subroutine begun at line 2900.

3000 This is the 'target' of the GOSUB in line 40

3001 The REMark \*\*\* PLAY is used to inform the reader of the main function of the subroutine starting at line 3000.

3010 through 3070 instruct the computer to perform various subroutines. These subroutines determine which format the story will reflect.

3075 and 3077 A legal filler and an empty line.

3080 The computer waits for you to indicate that you have finished reading the story by pressing RETURN. You could change this line to allow the reader to escape from further Sci-Fi literature.

```
3080 INPUT "HAD ENOUGH ";ANS$
3085 IF LEFT$(ANS$,1) = "Y" THEN END
```

Using the LEFT string function to find the first letter of "yes" is standard practice, and most computerists are used to answering 'Y' and expecting it to work. 'Yo', 'Yea', 'Yes', will all work. Any other response will end the program. Instead of END you could branch to a line at the end of the listings and add any farewell sequence you wished.

3100 This line first clears the screen of text (HOME). Then the function defined at line 2005 yields a RaNDom number between one and five. If the integer generated is one, then the program will branch to 3110. If FN R(5) generates the integer '2', then the program branches to 3120. If FN R(5) yields 3,4, or 5, then the program will branch to 3130,3140, or 3150 respectively.

3110 through 3150 These lines PRINT the five different attention-getting headlines.

3200 Line 2016 counted the total number of PLaces saved in PL\$. FN R(PL) will yield a RaNDom integer ranging from one to PL. There is the name of a different place saved in PL\$(1), PL\$(2), PL\$(3).....PL\$(PL). What this line does is to randomly select one of the many PLaces and to copy it into the location WRD\$. Then the subroutine beginning at line 2900 checks to make sure that the PLace is not PRINTEd in such a manner that it breaches the right margin.

3300 Like line 3100, selects a RaNDomly generated integer ranging from one to five (FN R(5)), and depending on the integer, branches to line 3310, 3320, 3330, 3340, or 3350.

3310 through 3350 These lines serve as continuations of the currently unfolding drama. One of the five rather repugnant actions is stored in the location WRD\$.

3400 Location WRD\$ already contains one of the five actions outlined on lines 3310 to 3350. These lines add the name of one of the many (MO) monsters (MO\$) to WRD\$. Line 2217 counted the number of monsters and stored the name of each one in an MO\$ location. This line RaNDomly selects one of the MOnters and adds its name, plus a trailing blank space (" ") to WRD\$. Each time the subroutine at 2900 is performed, the contents of WRD\$ are PRINTEd and cleared from WRD\$. Each time WRD\$ is emptied, the story grows.

3500 First, the word 'FROM' and a trailing space (" ") are stored in WRD\$. This word is added to the developing story. At the subroutine beginning at line 2900, the contents of WRD\$ (FROM ) are added to the rapidly developing story. Then, the HOme base of the MOnters is added to WRD\$, and a period (".") is also added. Once again WRD\$ is emptied (in the 2900 subroutine) and the story sprouts another section.

3600 From the list of potential heroes, SOmeone will be RaNDomly selected (FN R(SO)). A space (" ") will follow SOmeone's name, and then this material will be added to the story.

3700 First, the words "TRIED TO" are added to the text. Next, one of the five 'defense methods' is randomly selected and added to WRD\$ (at 3710, 3720, 3730, 3740, and 3750). Finally, the subroutine which begins at line 2900 adds the 'defense technique' to the story.

3710 through 3750 These lines contain the five 'defense techniques'.

3800 Two more words and a trailing space (BUT THEY) are added to the text. Next, one of five responses to SOmeone's 'defense technique' is RaNDomly chosen and the text is once again supplemented.

3810 through 3850 Each line contains one of the five responses to a 'defense technique'.

3900 There are five ways to announce the changing fortunes of battle. One of these five choices is randomly selected (FN R(5) GOTO....). The subroutine at



line 2900 again prevents right-hand margin overflow while adding the new material to the story.

3910 through 3950 These lines contain the five 'fortune-reversal lead-ins' mentioned at line 3900.

4000 You were given an explanation of the "ON" statement in line 1050, but it deserves reiteration. First, line 2005 DEFined the FuNction R(X) to be equal to a RaNDom INTegeR between 1 and X. The number 5 is substituted for 'X', making R(5) yield a random integer between one and five. The "ON" statement will cause the program to execute one of the five given subroutines, depending on the value of FN R(5). If the number generated is 3, then line 4000 reads, in effect, ON 3 GOSUB (the 3rd line-number) which is line... 4030. After one of the five subroutines is performed, the program executes the subroutine beginning at line 2900. Then RETURN completes the subroutine initiated at line 3060.

4010 through 4050 These five lines contain the different 'attack methods' that SOmeone might employ. The information is stored in WRD\$ until it is PRINTEd (by line 2900).

4100 A two word phrase, "SO THEY ", is stored in WRD\$ for later addition to the story. Next, the subroutine beginning at line 2900 is executed. This subroutine not only PRINTs the contents of WRD\$, but it prevents the contents of WRD\$ from being printed so that they breach the right margin. After one of the five 'attack results' is added to WRD\$, the subroutine which begins at line 2900 is executed. For all intents and purposes, the RETURN statement at line 4100 completes the program.

4110 through 4150 These lines contain the five 'results' of the 'attack method'.

Even as you read these words some of the illustrious characters whose names are stored in the data statements are slipping further into obscurity. Put in your own selection of names. This program is really an easy one to change and make your own. Change the data statements and print commands to your own fiendish specifications. You can save the new version under a new name and have both versions on diskette. Remember! What you change in the computer's memory does not change what is stored on the diskette unless you save the new version and give it the same name as the old version. It is sound practice to leave the original version unchanged on a different diskette and call your version Sci-Fi V1, Sci-Fi V2, etc. Have fun changing things around. It's the best way to learn.

```

10 REM *****
11 REM *** ***
12 REM *** SCI-FI ***
13 REM *** ***
14 REM *****
15 REM
16 REM
20 GOSUB 1000: REM INSTS
30 GOSUB 2000: REM SETUP
40 GOSUB 3000: REM PLAY!
50 END
1000
1001 REM *** INSTS
1002
1010 TEXT : NORMAL : HOME
1020 VTAB 3: HTAB 13: PRINT "***
    SCI-FI ***"
1030 VTAB 7: PRINT "THIS PROGRAM
    WILL PRODUCE LOTS OF FUNNY
    LITTLE SCIENCE-FICTION STORI
    ES FOR YOUR READING PLEASURE
    "
1040 PRINT : PRINT "YOU ARE GIVE
    N THE CHANCE TO ENTER SOME
    PERSONALLY RELEVANT INFORMAT
    ION IN HOPESOF MAKING THE ST
    ORIES MORE INDIVIDUAL."
1050 VTAB 23: INPUT "HIT RETURN
    WHEN READY TO CONTINUE : ";A
    NS$
1990 RETURN
2000
2001 REM *** SETUP
2002
2005 DEF FN R(X) = INT ( RND (
    1) * X) + 1
2010 DIM SO$(15):SO = 0
2011 DIM PL$(15):PL = 0
2012 DIM MO$(15):MO = 0
2013 DIM HO$(15):HO = 0
2015 READ SO$: IF SO$ < > "END"
    THEN SO = SO + 1:SO$(SO) =
    SO$: GOTO 2015

```

```

2016  READ PL$: IF PL$ < > "END"
      THEN PL = PL + 1:PL$(PL) =
      PL$: GOTO 2016
2017  READ MO$: IF MO$ < > "END"
      THEN MO = MO + 1:MO$(MO) =
      MO$: GOTO 2017
2018  READ HO$: IF HO$ < > "END"
      THEN HO = HO + 1:HO$(HO) =
      HO$: GOTO 2018
2020  DATA  "ALEXANDER HAIG","RON
      ALD REAGAN","SUPER CHICKEN",
      "FATHER GUIDO SARDUCCI","A L
      ITTLE GIRL","AN INTERESTED O
      NLOOKER","SUPERMAN"
2021  DATA  "END"
2025  DATA  "SYRACUSE","NEW YORK",
      "EARTH","THE UNITED STATES",
      "YOUR TOWN","LOS ANGELES","W
      ALLA-WALLA WASHINGTON","THE PR
      ESIDENT","THE EASTERN COAST"

2026  DATA  END
2030  DATA  "LITTLE GREEN MEN","A
      CROWD OF ANGRY PEASANTS BEA
      RING TORCHES","BOOGEY MEN","
      ALIENS","IN-LAWS","SPACE EGG
      S","FLYING GOOKIES","ZOMBIES
      ","RELIGIOUS FANATICS","ICKY
      THINGS"
2031  DATA  "END"
2035  DATA  "VENUS","MARS","OUTER
      SPACE","OUT OF OUR GALAXY","
      THE MOON","THE FOURTH DIMENS
      ION","THE NEGATIVE ZONE","A
      TIME WARP","THE STARS","PLUT
      O"
2036  DATA  "END"
2100  HOME : VTAB 3: HTAB 13: PRINT
      "*** SCI-FI ***": VTAB 22: PRINT
      "PRESS RETURN AT ANY TIME."
2110  VTAB 7: PRINT "TYPE IN UP T
      O 5 NAMES OF PEOPLE THAT
      WILL SAVE THE DAY : ": PRINT

```

```

2114 LET CO = 0
2115 INPUT "===> ";SO$: IF SO$ <
    > "" THEN SO = SO + 1:SO$(S
    0) = SO$:CO = CO + 1: IF CO <
    5 THEN 2115
2117 HOME : VTAB 3: HTAB 13: PRINT
    "*** SCI-FI ***": VTAB 7
2120 PRINT "TYPE IN UP TO 5 NAME
    S OF PLACES THAT WILL BE
    ATTACKED : ": PRINT
2124 LET CO = 0
2125 INPUT "===> ";PL$: IF PL$ <
    > "" THEN PL = PL + 1:PL$(P
    L) = PL$:CO = CO + 1: IF CO <
    5 THEN 2125
2127 HOME : VTAB 3: HTAB 13: PRINT
    "*** SCI-FI ***": VTAB 7
2130 PRINT "TYPE IN UP TO 5 NAME
    S OF MONSTERS THAT WILL ATT
    ACK : ": PRINT
2134 LET CO = 0
2135 INPUT "===> ";MO$: IF MO$ <
    > "" THEN MO = MO + 1:MO$(M
    0) = MO$:CO = CO + 1: IF CO <
    5 THEN 2135
2137 HOME : VTAB 3: HTAB 13: PRINT
    "*** SCI-FI ***": VTAB 7
2140 PRINT "TYPE IN UP TO 5 NAME
    S OF PLACES THAT THE MONSTERS
    COME FROM : ": PRINT
2144 LET CO = 0
2145 INPUT "===> ";HO$: IF HO$ <
    > "" THEN HO = HO + 1:HO$(H
    0) = HO$:CO = CO + 1: IF CO <
    5 THEN 2145
2150 RETURN
2900 LET PT = 1
2905 PRINT MID$ (WRD$,PT,1);: IF
    MID$ (WRD$,PT,1) = " " AND
    PEEK (36) > 30 THEN PRINT

2910 PT = PT + 1: IF PT < = LEN
    (WRD$) THEN 2905

```

```

2990 RETURN
3000 :
3001 REM *** PLAY
3002 :
3010 GOSUB 3100: REM TITLE
3015 GOSUB 3200: REM PLACE
3020 GOSUB 3300: REM ACTION
3025 GOSUB 3400: REM MONSTER
3030 GOSUB 3500: REM PLACE
3035 GOSUB 3600: REM SOMEONE
3040 GOSUB 3700: REM DEFEND
3045 GOSUB 3800: REM TOO TOUGH
3050 GOSUB 3900: REM FINALLY
3055 GOSUB 3600: REM SOMEONE
3060 GOSUB 4000: REM DEFEND
3070 GOSUB 4100: REM THEY DIED
3075 :
3077 PRINT
3080 VTAB 23: INPUT "HIT RETURN
      WHEN READY TO CONTINUE : ";A
      NS$
3085 GOTO 3010
3100 HOME : ON FN R(5) GOTO 311
      0,3120,3130,3140,3150
3110 VTAB 3: PRINT "*** FLASH! F
      LASH! FLASH! ***": VTAB 7: RETURN

3120 VTAB 3: PRINT "*** BULLETIN
      !!! ***": VTAB 7: RETURN
3130 VTAB 3: PRINT "*** ALERT !!
      ! ***": VTAB 7: RETURN
3140 VTAB 3: PRINT "*** SPECIAL
      NEWS BULLETIN ***": VTAB 7: RETURN
3150 VTAB 3: PRINT "*** TO ALL C
      ITIZENS ***": VTAB 7: RETURN

3200 WRD$ = PL$( FN R(PL)) + " ":
      GOSUB 2900: RETURN
3300 ON FN R(5) GOSUB 3310,3320
      ,3330,3340,3350: GOSUB 2900:
      RETURN
3310 WRD$ = "WAS ATTACKED BY ": RETURN

```

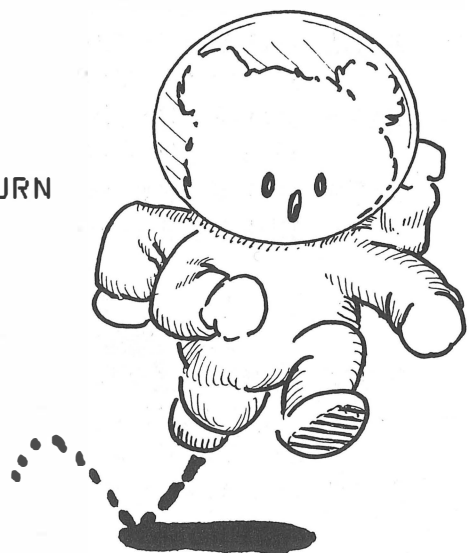
```

3320 WRD$ = "WAS EATEN BY ": RETURN
3330 WRD$ = "IS UNDER THE SPELL O
      F ": RETURN
3340 WRD$ = "IS BEING INVADED BY
      ": RETURN
3350 WRD$ = "IS OVER-RUN BY ": RETURN

3400 WRD$ = MO$( FN R(MO)) + " ":
      GOSUB 2900: RETURN
3500 WRD$ = "FROM ": GOSUB 2900:W
      RD$ = HO$( FN R(HO)) + ", ":
      GOSUB 2900: RETURN
3600 WRD$ = SO$( FN R(SO)) + " ":
      GOSUB 2900: RETURN
3700 WRD$ = "TRIED TO ": GOSUB 29
      00: ON FN R(5) GOSUB 3710,3
      720,3730,3740,3750: GOSUB 29
      00: RETURN
3710 WRD$ = "KILL THEM ": RETURN
3720 WRD$ = "FIGHT THEM ": RETURN
3730 WRD$ = "HOLD UP A CROSS ": RETURN
3740 WRD$ = "ATTACK AT DAWN ": RETURN
3750 WRD$ = "SHOOT THEM ": RETURN

3800 WRD$ = "BUT THEY ": GOSUB 29
      00: ON FN R(5) GOSUB 3810,3
      820,3830,3840,3850: GOSUB 29
      00: RETURN
3810 WRD$ = "WERE TOO TOUGH. ": RETURN
3820 WRD$ = "KEPT COMING. ": RETURN
3830 WRD$ = "YELLED AND LAUGHED,
      ": RETURN
3840 WRD$ = "SCREAMED FOR MORE. "
      : RETURN
3850 WRD$ = "SHOT BACK. ": RETURN

```





```
3900  ON  FN R(5) GOSUB 3910,3920
      ,3930,3940,3950: GOSUB 2900:
      RETURN
3910  WRD$ = "FINALLY, ": RETURN
3920  WRD$ = "LATER, ": RETURN
3930  WRD$ = "THEN ., . ": RETURN
3940  WRD$ = "BUT THEN, ": RETURN


3950  WRD$ = "AFTER, ": RETURN
4000  ON  FN R(5) GOSUB 4010,4020
      ,4030,4040,4050: GOSUB 2900:
      RETURN
4010  WRD$ = "YELLED AT THEM, ": RETURN

4020  WRD$ = "DROPPED WATER ON THE
      M, ": RETURN
4030  WRD$ = "EXPOSED THEM TO MEAS
      LES ": RETURN
4040  WRD$ = "NUKED THEM, ": RETURN

4050  WRD$ = "SHOWED THEM RERUNS O
      F I LOVE LUCY, ": RETURN
4100  WRD$ = "SO THEY ": GOSUB 290
      0: ON  FN R(5) GOSUB 4110,41
      20,4130,4140,4150: GOSUB 290
      0: RETURN
4110  WRD$ = "DIED.": RETURN
4120  WRD$ = "TURNED INTO LITTLE B
      ROWN LUMPS.": RETURN
4130  WRD$ = "PASSED AWAY.": RETURN

4140  WRD$ = "LEFT FOR HOME.": RETURN

4150  WRD$ = "VANISHED INTO NOTHIN
      GNESS.": RETURN
```



This game is designed to compel the user to define any object that is chosen. This process 'teaches' the computer a definition of up to fifty objects. Although each run starts with the same elementary knowledge, the user adds the information which makes it harder and harder to stump the computer. To see the "setup," list -2030. The only two objects which the computer "knows" are a car and a house. Line 2015 specifies that RA\$(1) = car, and WA\$(1) = house. Line 2010 gives each of these variables fifty locations, so you can play for a long time. What the DIM statement does is to DIMension memory so that BA\$, WA\$, etc. will be able to contain up to fifty separate values. In other words, BA\$(1), BA\$(2), BA\$(3) ....BA\$(50) each contain a distinct value. This program demonstrates writing to array tables and searching the arrays for matching strings. You could use the routines in this program to write educational programs and tests.

```

10 REM *****
11 REM *** ***
12 REM ** APPLE LEARNER **
13 REM *** ***
14 REM *****
15 REM
16 REM
20 GOSUB 1000: REM INSTS
30 GOSUB 2000: REM SETUP
40 GOSUB 3000: REM PLAY!
50 END
1000
1001 REM *** INSTS
1002
1010 TEXT : HOME : NORMAL
1020 VTAB 3: HTAB 9: PRINT "***
APPLE LEARNER ***"
1030 VTAB 7: PRINT "THIS IS A GA
ME THAT HAS THE ABILITY TO
LEARN. IT WILL ATTEMPT TO G
UESS THE NAME OF AN OBJEC
T THAT YOU PICK AT RAND
OM."
1040 PRINT
1041 PRINT "WHENEVER YOU STUMP T
HE COMPUTER, YOU AREASKED AB
OUT THE OBJECT YOU SELECTED.
BYCOMPILING THIS INFORMATI
ON, THE COMPUTER'LEARNS'."
1045 PRINT : PRINT
1046 PRINT "ENTER 'STOP' WHEN YO
U ARE DONE. "
1050 VTAB 23
1051 INPUT "HIT RETURN WHEN READ
Y TO CONTINUE : ";ANS$
1990 RETURN
2000
2001 REM *** SETUP
2002
2010 DIM QU$(50),RI(50),WR(50),R
A$(50),WA$(50)
2015 QU$(1) = "DOES IT MOVE ALONG
THE GROUND":RI(1) = 0:WR(1)
= 0

```

```

2020 RA$(1) = "CAR":WA$(1) = "HOU
      SE"
2030 FR = 2
2990 RETURN
3000 :
3001 REM *** PLAY
3002 :
3005 LI = 1
3010 HOME : VTAB 3: HTAB 9: PRINT
      "*** APPLE LEARNER ***": VTAB
3015 PRINT "I KNOW OF "FR" OBJEC
      TS ...": PRINT
3020 PRINT : PRINT QU$(LI);: INPUT
      " ? ";ANS$:ANS$ = LEFT$ (AN
      S$,1)
3030 IF LEFT$ (ANS$,1) = "Y" THEN
      3100
3035 IF LEFT$ (ANS$,1) = "N" THEN
      3200
3036 IF LEFT$ (ANS$,1) = "S" THEN
      TEXT : END
3040 PRINT "PLEASE ANSWER 'YES'
      OR 'NO'...": PRINT : GOTO 30
      20
3100 IF RI(LI) THEN LI = RI(LI):
      GOTO 3020
3105 GU$ = RA$(LI): GOTO 3300
3200 IF WR(LI) THEN LI = WR(LI):
      GOTO 3020
3205 GU$ = WA$(LI): GOTO 3300
3300 PRINT "IS IT A ";GU$;
3310 INPUT " ? ";TA$:TA$ = LEFT$
      (TA$,1): IF TA$ = "Y" THEN PRINT
      : PRINT "I GOT IT !!!": CHR$
      (7); CHR$ (7); CHR$ (7): FOR
      PA = 1 TO 1000: NEXT PA: GOTO
      3005
3312 IF TA$ = "S" THEN TEXT : END

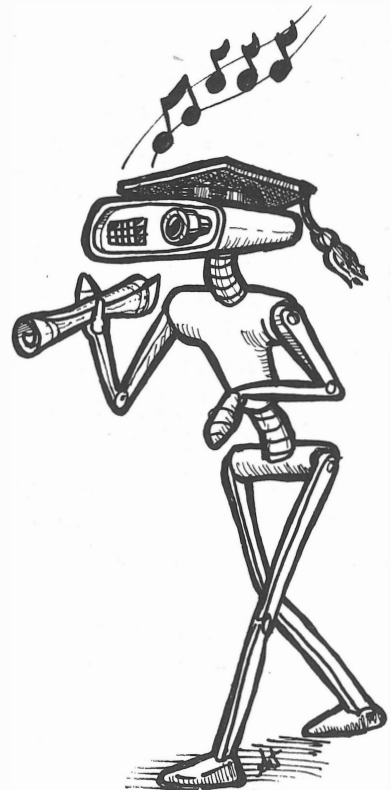
3315 PRINT : PRINT : INPUT "WHAT
      WAS THE OBJECT?";NA$

```

```

3317 IF FR = 51 THEN PRINT "I C
      AN'T REMEMBER THAT ONE. MY
      MEMORY SEEMS TO BE FULL...
      ": FOR PA = 1 TO 1000: NEXT
      PA: GOTO 3005
3320 PRINT : PRINT "WHAT IS A QU
      ESTION THAT I COULD USE TO
      TELL THE DIFFERENCE BETWEEN
      ": PRINT GU$ AND "NA$"; INPUT
      " ? "; QU$
3325 PRINT "FOR "NA$" THE ANSWER
      IS WHAT"; INPUT " ? "; YN$:
      YN$ = LEFT$(YN$,1): IF YN$
      < > "Y" AND YN$ < > "N
      " THEN 3325
3340 IF AN$ = "Y" THEN RI(LI) =
      FR:LI = FR:FR = FR + 1
3341 IF AN$ = "N" THEN WR(LI) =
      FR:LI = FR:FR = FR + 1
3345 QU$(LI) = QU$
3350 IF YN$ = "Y" THEN RA$(LI) =
      NA$:WA$(LI) = GU$
3351 IF YN$ = "N" THEN RA$(LI) =
      GU$:WA$(LI) = NA$
3355 GOTO 3005

```



This program is designed to interpret input, and from it, graph the user's biorhythms. You can accept the output to be as valid as you please, but don't expect the results to be testimony. Biorhythms, though fascinating, are still considered to be unscientific. There are several features of the program worth noting even if it won't predict the future:

Lines 2010 to 2025 set up a simple calendar printing routine. A few more lines and you could add leap years to it.

Line 3020 asks for a date and tells you exactly how to format the response.

Lines 3025-3028 'error check' the input. Error checking is a vital part of so-called user friendly programs.

Line 3140-3148 position the P, E and C characters into beautiful sine curves. Even if you have forgotten all your high school math you can try out the trig functions by plotting points to give a graphic representation of the function.

Did you recognize 6.28318 as  $2\pi$ ?

Running BIORHYTHM in the TRACE mode will show you very dramatically the FOR-NEXT loops in 3152 and 3208 that create the patterns.

The program uses complicated string functions and nested loops, so don't feel bad if it all looks like Relativity Theory. You can write a lot of programs which never get this complex.



```

10 REM *****
11 REM *** ***
12 REM *** BIORHYTHM ***
13 REM *** ***
14 REM *****
15 REM
16 REM
18 GOSUB 45
20 GOSUB 1000: REM INSTS
30 GOSUB 2000: REM SETUP
40 GOSUB 3000: REM PLAY!
42 END
45 HOME : VTAB 4
46 PRINT "A GRAPH OF BIORHYTHMS
    IS COMPOSED OF THE LETTERS C,
    E, I, AND P. EACH OF THESE
    IS";
47 PRINT " A REPRESENTATION OF ONE OF YOUR ";
48 PRINT "MEASURABLE BIORHYTHMS.
    "
50 FOR I = 1 TO 4: PRINT : NEXT
    : HTAB 6
51 PRINT "I = INTELLECTUAL STATE
    "
52 PRINT : HTAB 6
53 PRINT "E = EMOTIONAL STATE"
54 PRINT : HTAB 6
55 PRINT "P = PHYSICAL STATE"
56 PRINT : HTAB 6
57 PRINT "C = THE CROSSOVER POINT"
58 FOR I = 1 TO 6: PRINT : NEXT
    : INPUT "PRESS RETURN WHEN READY TO CONTINUE : "; AN$
70 RETURN
1000 :
1001 REM *** INSTS
1002 :
1010 TEXT : NORMAL : HOME
1020 VTAB 3: HTAB 11: PRINT "***
    BIORHYTHM ***"
1030 VTAB 7

```

```

1031 PRINT "THIS PROGRAM WILL GR
APH OUT YOUR UNIQUE BIORHYTH
MIC CYCLES, EITHER ON THE SC
REEN OR TO A PRINTER.  "
1040 VTAB 16: PRINT "SHOULD I OU
TPUT TO:
1042 PRINT PRINT "          S)CR
EEN
          -OR-
1045 PRINT "          P)RINTER
1050 VTAB 22
1051 INPUT "WHICH DO YOU WANT (S
/P) ? ";ANS$:ANS$ = LEFT$ (
ANS$,1): IF ANS$ < > "S" AND
ANS$ < > "P" THEN 1050
1060 IF ANS$ = "S" THEN RETURN

1065 VTAB 22: CALL - 958: INPUT
"IN WHICH SLOT IS YOUR PRINT
ER?";SLOT$:SLOT = VAL (SLOT
$): IF SLOT < 1 OR SLOT > 7 OR
SLOT < > INT (SLOT) THEN 1
065
1990 RETURN
2000
2001 REM *** SETUP
2002
2010 DIM A(12),B(12),T(3),A$(21)

2015 C$ = "JANFEBMARAPRMAYJUNJULA
UGSEPOCTNOVDEC"
2020 FOR I = 1 TO 12: READ A(I):
NEXT : DATA 0,31,59,90,120,
151,181,212,243,273,304,334
2025 FOR I = 1 TO 12: READ B(I):
NEXT : DATA 31,28,31,30,31,
30,31,31,30,31,30,31
2990 RETURN
3000
3001 REM *** PLAY
3002
3010 HOME : VTAB 3: HTAB 11: PRINT
"*** BIORHYTHM ***"

```

```

3015 INPUT "WHAT IS YOUR NAME? "
    ;N$
3020 PRINT : PRINT "WHAT IS YOUR
    BIRTHDATE? ": INPUT "MM,DD,
    YYYY) ";M,D,Y
3025 IF M < 1 OR M > 12 THEN PRINT
    "INCORRECT MONTH":ER = 1
3026 IF D < 1 OR D > 31 THEN PRINT
    "INCORRECT DAY":ER = 1
3027 IF Y < 1900 OR Y > 1999 THEN
    PRINT "INCORRECT YEAR":ER =
    1
3028 IF ER THEN ER = 0: GOTO 302
    0
3030 PRINT : PRINT "WHAT IS THE
    START DATE?": INPUT "MM,DD,Y
    YY ";M1,D0,Y1
3035 IF M1 < 1 OR M1 > 12 THEN PRINT
    "INCORRECT MONTH":ER = 1
3036 IF D0 < 1 OR D0 > 31 THEN PRINT
    "INCORRECT DAY":ER = 1
3037 IF Y1 < 1900 THEN PRINT "I
    NCORRECT YEAR":ER = 1
3038 IF ER THEN ER = 0: GOTO 303
    0
3040 PRINT : INPUT "HOW MANY DAY
    S? ";Z: IF Z < 1 OR Z > INT
    (Z) THEN 3040
3045 W = D0:W1 = M1:W2 = Y1:W3 =
    Z
3050 J = A(M) + D:D1 = 365 - J +
    ((J < = 60) AND (Y / 4 = INT
    (Y / 4))):D2 = 365 * (Y1 - (
    Y + 1)):E = 0: FOR T = Y + 1
    TO Y1 - 1:E = E + (T / 4 =
    INT (T / 4)): NEXT T
3055 D3 = A(M1) + D0:D3 = D3 + ((
    Y / 4 = INT (Y / 4)) AND (D
    3 > = 60)):D4 = D1 + D2 + D
    3 + E
3056 IF D4 < 0 THEN PRINT "STAR
    T DATE BEFORE BIRTH DATE": GOTO
    3030

```

```

3104 P1 = D4      INT (D4 / 23) * 2
      3
3108 E1 = D4 -   INT (D4 / 28) * 2
      8
3112 C1 = D4      INT (D4 / 33) * 3
      3
3116 IF ANS$ = "P" THEN PRINT CHR$
      (4)"PR#"SLOT: PRINT CHR$ (9
      )"80N"
3117 PRINT TAB( 22)"BIORHYTHM C
      YCLES": PRINT TAB( 25)"---
      FOR ---": PRINT TAB( 22 + (
      17 - LEN (N$)) / 2)N$
3118 PRINT TAB( 25); MID$ (C$,3
      * M - 2,3)" "D", "Y: PRINT

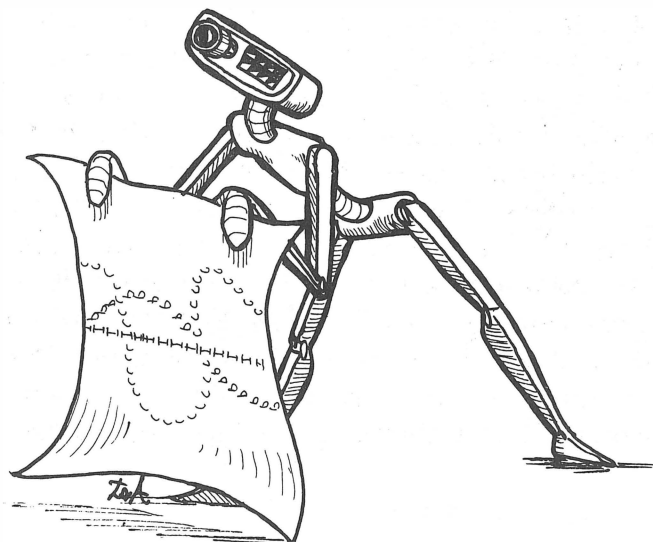
3120 PRINT MID$ (C$,3 * M1 - 2,
      3)" "Y1"          (-)
      0)      (+)"
3124 FOR T = 1 TO Z
3128 P2 = P1 + T - INT ((P1 + T)
      / 23) * 23
3132 E2 = E1 + T - INT ((E1 + T)
      / 28) * 28
3136 C2 = C1 + T - INT ((C1 + T)
      / 33) * 33
3140 P3 = INT (11.5 + 10 * SIN
      (P2 * 6.28318 / 23))
3144 E3 = INT (11.5 + 10 * SIN
      (E2 * 6.28318 / 28))
3148 C3 = INT (11.5 + 10 * SIN
      (C2 * 6.28318 / 33))
3152 FOR I = 1 TO 21:A$(I) = " "
      : NEXT
3156 A$(P3) = "P"
3160 IF A$(E3) < > " " THEN 318
      0
3164 A$(E3) = "E"
3168 IF A$(C3) < > " " THEN 318
      8
3172 A$(C3) = "C": GOTO 3192
3180 A$(E3) = "*": GOTO 3168
3188 A$(C3) = "*": GOTO 3192

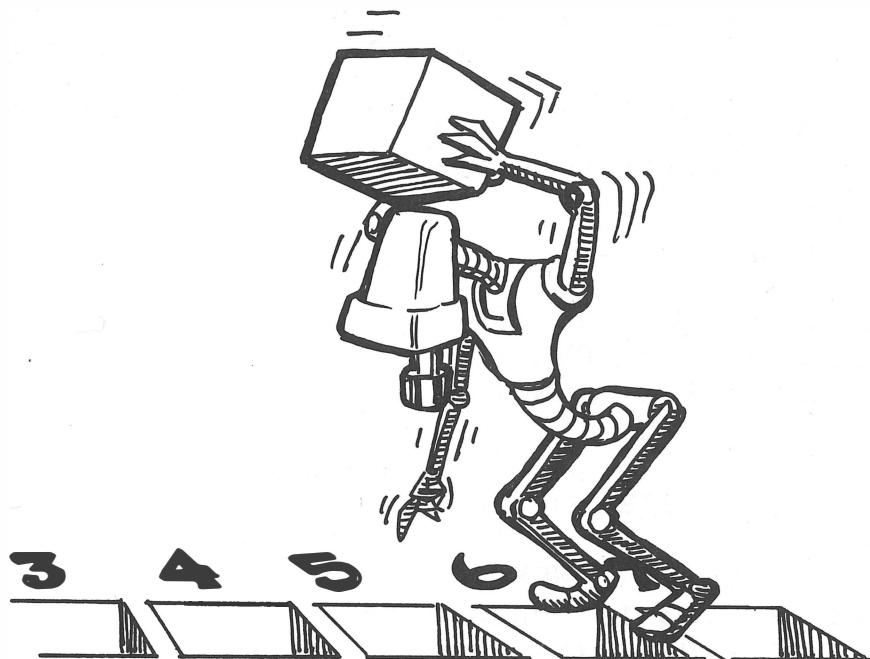
```

```

3192 IF A$(11) = " " THEN A$(11)
    = "I"
3196 IF D0 = 1 THEN PRINT MID$
    (C$,3 * M1 - 2,3) " "; GOTO
    3208
3198 PRINT " ";
3208 PRINT RIGHT$ (" " + STR$
    (D0),2) " "; FOR
    I = 1 TO 21: PRINT A$(I);: NEXT
    : PRINT
3212 IF Y1 - (( INT (Y1 / 4)) *
    4) = 0 THEN B(2) = 29
3224 D0 = D0 + 1: IF D0 > B(M1) THEN
    D0 = 1:M1 = M1 + 1
3244 IF M1 > 12 THEN M1 = 1:Y1 =
    Y1 + 1
3260 NEXT T
3300 IF ANS$ = "P" THEN PRINT CHR$
    (4)"PR#0"
3990 RETURN

```





# Connect Five

This classic game requires that you connect five squares either vertically, horizontally, or diagonally. Though not much of a challenge, the game is good for your ego. The graphics are fairly basic, so let's take a closer look. First, type: LIST-2025. Experiment with the color, the line length and location, and the FOR statement. After you are done analyzing those lines, type: LIST -3050. Most noteworthy are lines 3010-3021. Line 3010 asks you to input the desired column NUMBER, but the variable (ANS\$) is for numbers and characters. Any variable that ends with a dollar-sign (\$) is called a string and is not capable of having any mathematical functions performed upon it. The reason the variable in 3010 is a string variable is that the person choosing the column number might accidentally hit a letter instead of a number. If the variable in 3010 was ANS, and an I were input in place of 1, the program would 'crash'. Each character has a corresponding numeric value, so a letter can be redefined as a numeric. Line 3020 converts the string ANS\$ to the numeric ANS. Line 3021 then makes mathematical comparisons based upon the input. You might want to experiment with the VAL command to see how computers alphabetize lists of words.

```

10  REM *****
11  REM *** ***
12  REM *** CONNECT FIVE ***
13  REM *** ***
14  REM *****
15  REM
16  REM
20  GOSUB 1000: REM  INSTS
30  GOSUB 2000: REM  SETUP
40  GOSUB 3000: REM  PLAY!
50  GOSUB 4000: REM  !END!
60  END
1000 :
1001  REM  *** INSTS
1002 :
1010  TEXT : NORMAL : HOME
1020  VTAB 3: HTAB 10: PRINT "***
      CONNECT FIVE ***"
1030  VTAB 7: PRINT "THE OBJECT O
      F THE GAME IS TO GET FIVE OF
      YOUR PIECES IN A ROW, EITHER
      VERTICALLY OR HORIZONTALLY."
1035  PRINT
1040  PRINT "WHEN IT IS YOUR TURN
      TO MOVE, ENTER THE NUMBER O
      F THE COLUMN YOU WISH TO DRO
      P  YOUR PIECE INTO, "
1045  PRINT
1050  PRINT "AFTER YOU MOVE, I WI
      LL TAKE A TURN,      THE FIRS
      T ONE TO CONNECT FIVE IS THE
      WINNER, "
1060  VTAB 22: INPUT "PRESS RETUR
      N WHEN READY TO CONTINUE : "
      ;ANS$
1990  RETURN
2000 :
2001  REM  *** SETUP
2002 :
2010  GR
2011  COLOR= 0
2012  FOR I = 0 TO 35
2013  HLIN 0,35 AT I
2014  NEXT

```



```

2015  COLOR= 15
2016  FOR I = 0 TO 35 STEP 5
2017  HLIN 0,35 AT I: VLIN 0,35 AT
    I
2018  NEXT I
2020  COLOR= 6
2021  VLIN 35,39 AT 3
2022  HLIN 2,4 AT 39
2023  PLOT 2,36
2024  HLIN 7,9 AT 35: HLIN 7,9 AT
    37: HLIN 7,9 AT 39
2025  PLOT 9,36: PLOT 7,38
2030  HLIN 12,14 AT 35: HLIN 12,1
    4 AT 37: HLIN 12,14 AT 39: PLOT
    14,36: PLOT 14,38: HLIN 17,1
    9 AT 37: VLIN 35,39 AT 19: VLIN
    35,36 AT 17
2040  HLIN 22,24 AT 35: HLIN 22,2
    4 AT 37: HLIN 22,24 AT 39: PLOT
    22,36: PLOT 24,38: HLIN 27,2
    9 AT 35: HLIN 27,29 AT 37: HLIN
    27,29 AT 39
2050  PLOT 27,38: PLOT 29,38: PLOT
    27,36: HLIN 32,34 AT 35: PLOT
    34,36: VLIN 37,39 AT 33
2900  DEF FN C(X) = (X  1) * 5 +
    1
2910  DEF FN P(X) = (X  1) * 5 +
    9
2990  RETURN
3000
3001  REM  *** PLAY
3002
3010  HOME :PL = 1: INPUT "YOUR M
    OVE (COLUMN 1-7) : ";ANS$
3020  ANS = VAL (ANS$)
3021  IF ANS < 1 OR ANS > 7 OR AN
    S < > INT (ANS) THEN HOME
    : PRINT "PLEASE SELECT A NUM
    BER FROM 1 TO 7 : ": FOR A =
    1 TO 1000: NEXT A: GOTO 3010

3030  AX = FN C(ANS)

```

```

3040 IF SCRN( AX,1) < > 0 THEN
    HOME : PRINT "THAT COLUMN I
    S FULL... ": FOR A = 1 TO 10
    00: NEXT A: GOTO 3010
3050 FOR J = 1 TO 7: COLOR= FN
    P(PL):JX = FN C(J): FOR K =
    JX TO JX + 3: HLIN AX,AX + 3
    AT K: NEXT K
3055 IF J = 7 THEN 3080
3060 LX = FN C(J + 1): IF SCRN(
    AX,LX) < > 12 THEN J = 7: GOTO
    3080
3065 COLOR= 0
3070 FOR K = JX TO JX + 3: HLIN
    AX,AX + 3 AT K: NEXT K
3080 NEXT J
3090 FOR I = 1 TO 7: FOR J = 1 TO
    3: FOR K = J TO J + 4:IX = FN
    C(I):KX = FN C(K)
3100 IF SCRN( IX,KX) < > FN P
    (PL) THEN K = J + 4: NEXT K:
    GOTO 3110
3105 NEXT K: RETURN
3110 NEXT J,I
3120 FOR J = 1 TO 7: FOR I = 1 TO
    3: FOR K = I TO I + 4:KX = FN
    C(K):JX = FN C(J)
3130 IF SCRN( KX,JX) < > FN P
    (PL) THEN K = I + 4: NEXT K:
    GOTO 3140
3135 NEXT K: RETURN
3140 NEXT I,J
3150 IF PL = 2 THEN PL = 1: GOTO
    3010
3160 PL = 2:ANS = INT ( RND (1) *
    7) + 1:AX = FN C(ANS): IF SCRN(
    AX,1) < > 12 THEN 3160
3170 GOTO 3050
4000 :
4001 REM *** ALL DONE
4002 :

```

```
4010 HOME : PRINT : PRINT "THE G  
AME IS OVER !!!": PRINT "THE  
WINNER IS ... ";  
4020 IF PL = 1 THEN PRINT "YOU  
!!!": RETURN  
4030 PRINT "ME !!!": RETURN
```



This is definitely a thinking man's game. You are given clues in an attempt to guess a three-number puzzle. Load the program. FN R(10) generates a random integer between 0 and 9. How and why will be discussed elsewhere in the book, for now, just accept that this is true. N1 is any digit between 0 and 9. N2 is any digit between 0 and 9 except N2 cannot equal N1. N3 is also a number between 0 and 9. N3 cannot be equal to N2 or N1. The result is that the three-digit number represented by N1/N2/N3 will be a random three-digit number comprised of three different digits. Lines 3035-3037 separate your single three-digit guess into three separate guesses (G1, G2, and G3). Here's how. When any number is converted to an integer, the portion of the number which is to the right of the decimal point is truncated (cut off). Here are a few examples: INT 3.4 is 3, INT 9.989 is 9, INT 562.3 is 562, INT 0.3 is 0. Taking line 3035, assume that the guess was 567. 567 divided by 100 is 5.67. When converted to an INTeGer, 5.67 becomes 5 (G1 = 5). Line 3036 takes 567, subtracts (5 multiplied by 100), divides the result (67) by 10, and converts 6.7 into the INTeGer. Now G1 = 5 and G2 = 6. Lastly, 3037 takes 567 (ANS) and subtracts from it (5 (G1) times 100 plus 6 (G2) times 10) or 560. The result (567 - 560) is now stored in G3. So now G1 = 5, G2 = 6, and G3 = 7. Statistically, even when you are unlucky, the solution can be derived in no more than seven guesses. You've been challenged, now go to it!

```

10 REM *****
11 REM *** ***
12 REM *** DIGITS ***
13 REM *** ***
14 REM *****
15 REM
16 REM
20 GOSUB 1000: REM INSTS
30 GOSUB 2000: REM SETUP
40 GOSUB 3000: REM PLAY!
50 GOSUB 4000: REM !END!
60 END
1000 :
1001 REM *** INSTS
1002 :
1010 TEXT : NORMAL : HOME
1020 VTAB 2: HTAB 13: PRINT "***
DIGITS ***"
1030 VTAB 5
1031 PRINT "I WILL THINK OF A NU
MBER BETWEEN 012 AND987. EA
CH DIGIT IN THE NUMBER WILL
BE DIFFERENT FROM THE OTHER
TWO."
1035 PRINT
1040 PRINT "THE OBJECT OF THE GA
ME IS TO GUESS THE SOLUTION
IN AS FEW TRIES AS POSSIBLE
."
1050 VTAB 23: INPUT "HIT RETURN
WHEN READY TO CONTINUE : ";A
NS$
1060 HOME : VTAB 2: HTAB 13: PRINT
*** DIGITS ***: VTAB 5
1070 PRINT "AFTER EACH GUESS, I
WILL PRINT OUT A HINT LIN
E AS FOLLOWS : "
1075 PRINT
1080 PRINT ">FOR EACH DIGIT CORR
ECT AND IN THE CORRECT
POSITION, I WILL PRINT AN 'X
',"
1085 PRINT

```

```

1090 PRINT ">FOR EACH DIGIT CORR
      ECT BUT NOT IN THE CORRECT
      POSITION, I WILL PRINT AN 'O
      ', "
1095 PRINT
1100 PRINT ">FOR EACH TOTALLY IN
      CORRECT DIGIT, I WILL PRI
      NT A '-', "
1105 PRINT
1110 PRINT "PLAY WILL CONTINUE U
      NTIL YOU GUESS THE NUMBER,
      TO QUIT EARLY, SIMPLY HIT T
      HE RETURN KEY FOR YOUR GUES
      S."
1115 VTAB 23
1120 INPUT "HIT RETURN WHEN READ
      Y TO CONTINUE : ";ANS$
1130 HOME : VTAB 6
1140 PRINT "HERE IS A TABLE TO H
      ELF YOU UNDERSTAND THE INST
      RUCTIONS."
1145 PRINT : PRINT
1150 PRINT "ANSWER          GUESS
      HINT LINE "
1160 PRINT "-----"
      "-----"
1165 PRINT
1170 PRINT " 065          703
      0--
1171 PRINT
1175 PRINT " 562          463
      X--
1176 PRINT
1180 PRINT " 918          890
      00-": PRINT
1185 PRINT " 390          305
      X0-": PRINT
1190 PRINT " 271          721
      X00": PRINT
1195 PRINT " 425          780
      ---": PRINT
1200 INPUT "PRESS RETURN WHEN RE
      ADY TO CONTINUE : ";ANS$

```

```

1990 RETURN
2000 :
2001 REM *** SETUP
2002 :
2010 DEF FN R(X) = INT ( RND (
    1) * X)
2020 N1 = FN R(10)
2021 N2 = FN R(10): IF N2 = N1 THEN
    2021
2022 N3 = FN R(10): IF N3 = N1 OR
    N3 = N2 THEN 2022
2990 RETURN
3000 :
3001 REM *** PLAY
3002 :
3010 HOME : VTAB 3: HTAB 13: PRINT
    "*** DIGITS ***": VTAB 7
3020 PRINT "OKAY, I'VE GOT A NUM
    BER...": PRINT
3030 INPUT "WHAT IS YOUR GUESS?
    ";ANS$: IF ANS$ = "" THEN RETURN
3031 IF LEN (ANS$) < > 3 THEN
    PRINT "TYPE ONLY THREE DIGI
    TS PLEASE": GOTO 3030
3032 ANS = VAL (ANS$): IF ANS <
    0 OR ANS > 999 THEN PRINT "
    TYPE ONLY THREE DIGITS PLEASE"
    : GOTO 3030
3035 G1 = INT (ANS / 100)
3036 G2 = INT ((ANS - G1 * 100) /
    10)
3037 G3 = ANS - (G1 * 100 + G2 *
    10)
3040 IF G1 = G2 OR G1 = G3 OR G2
    = G3 THEN PRINT "TYPE THREE
    DIFFERENT DIGITS PLEASE.":
    GOTO 3030
3050 CP = 0:CD = 0:MI = 0
3051 IF G1 = N1 THEN CP = CP + 1
3052 IF G2 = N2 THEN CP = CP + 1
3053 IF G3 = N3 THEN CP = CP + 1

```



```

3055 IF G1 = N2 OR G1 = N3 THEN
    CD = CD + 1
3056 IF G2 = N1 OR G2 = N3 THEN
    CD = CD + 1
3057 IF G3 = N1 OR G3 = N2 THEN
    CD = CD + 1
3060 MI = 3 - CP - CD
3065 PRINT "FOR YOUR GUESS OF "G
    1;G2;G3", I HINT ";
3070 IF CP > 0 THEN FOR I = 1 TO
    CP: PRINT "X";: NEXT
3071 IF CD > 0 THEN FOR I = 1 TO
    CD: PRINT "O";: NEXT
3072 IF MI > 0 THEN FOR I = 1 TO
    MI: PRINT "-";: NEXT
3075 PRINT : PRINT :NG = NG + 1:
    IF CP = 3 THEN RETURN
3080 GOTO 3030
4000 :
4001 REM *** END
4002 :
4010 PRINT "THE GAME IS OVER ...
    ": PRINT
4015 IF CP = 3 THEN PRINT "YOU
    GUESSED IT IN ONLY "NG" TRIE
    S ! "
4020 IF CP < 3 THEN PRINT "THE
    CORRECT ANSWER WAS " N1;N2;N3
4990 RETURN

```





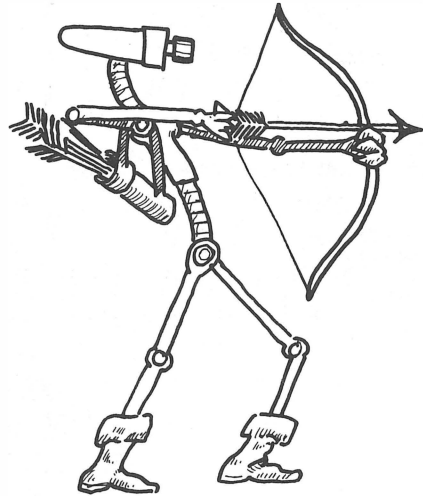
To play this game, you should have paper and pencil. You travel through an unseen maze of caverns searching for the Grue. As any good spelunker will tell you, drawing a map will prevent you from making the same mistakes over and over again. In other words, draw a map as you go along. There are no color graphics used in this program, but there are some other interesting features. You may ask, "How come I get different responses each time I run the program?" and here's why. Lines 2100 through 2130 assign certain variables a RANDOM value. Beginning at 3020, this becomes relevant. One of four messages is printed. Which one it is, depends on the RaNDom values of the four variables (EX, P1, B1, GU). Line 3026 reveals that on each move you have a one in fifteen chance of experiencing an earthquake. To paraphrase line 3026: if a RaNDom INTeger between 0 and 14 happens to be equal to 4, then PRINT (BELL\$ causes the computer to emit a ringing sound) <<< EARTHQUAKE >>>. Line 3030 really begins each turn. As you go through the program, you will note that the execution of most of the lines depends, either directly or indirectly, on the value of a randomly generated integer.

```

10 REM *****
11 REM *** ***
12 REM *** GRUE STEW ***
13 REM *** ***
14 REM *****
15 REM
16 REM
20 GOSUB 1000: REM INSTS
30 GOSUB 2000: REM SETUP
40 GOSUB 3000: REM PLAY!
50 GOSUB 4000: REM !END!
60 END
1000 :
1001 REM *** INSTS
1002 :
1010 TEXT : NORMAL : HOME
1020 VTAB 3: HTAB 12: PRINT "***
    GRUE STEW ***"
1030 VTAB 7: PRINT "IN THIS GAME
    , YOU ARE A BRAVE HUNTER.
    YOU ARE ALSO VERY HUNGRY. S
    O, YOU DECIDE TO GO 'GR
    UE' HUNTING. A GRUE, ASEVER
    YONE KNOWS, IS THE KEY INGRE
    DIENT INGRUE STEW."
1040 PRINT : PRINT "YOU ARE GOIN
    G TO ENTER A SERIES OF
    UNDERGROUND CAVES, IN SEARCH
    OF THE STEWBASE, THE GRUE."

1050 PRINT : PRINT "IF YOU CAN B
    AG A GRUE, AND GET OUT OF
    THE CAVES, THEN YOU WILL GET
    YOUR STEW (AND WIN THE GAM
    E!)."
1060 VTAB 23: INPUT "HIT RETURN
    WHEN READY TO CONTINUE : ";A
    NS$
1070 HOME : VTAB 3: HTAB 12: PRINT
    "*** GRUE STEW ***": VTAB 7

```



```

1080 PRINT "ONCE IN THE MAZE, YO
      U CAN EITHER MOVE TO A DIF
      FERENT CAVERN OR SHOOT AN AR
      ROW INTO AN ADJOINING CAVE,
      IN HOPES OF HITTING A FE
      ROCIOUS GRUE. "
1090 PRINT : PRINT "I WILL ASK:
      MOVE OR SHOOT?, AND YOU MUST
      REPLY WITH 'M' FOR MOVE OR '
      S' FOR SHOOT."
1095 PRINT : PRINT "IF YOU DECID
      E TO MOVE, YOU CAN DO SO IN
      ANY OF THE FOUR COMPASS DIRE
      CTIONS. WHEN ASKED WHICH WAY,
      ENTER 'N' FOR NORTH, 'S'
      FOR SOUTH, 'W' FOR WEST, OR
      'E' FOR EAST."
1100 VTAB 23: INPUT "HIT RETURN
      WHEN READY TO CONTINUE : ";A
      NS$
1120 HOME : VTAB 3: HTAB 12: PRINT
      "*** GRUE STEW ***": VTAB 7
1130 PRINT "IF YOU DECIDE TO SHO
      OT, YOU WILL BE ASKED: S
      HOOT WHICH WAY?, AND YOU MUS
      T REPLY: 'N' FOR NORTH, 'S
      ' FOR SOUTH, 'E' FOR EAST, 'W
      ' FOR WEST."
1140 PRINT : PRINT "IF YOU HIT T
      HE GRUE, YOU WILL BE TOLD,
      AND YOU MUST TRY TO EXIT THE
      CAVES."
1150 PRINT : PRINT "BUT...THERE
      ARE OTHER THINGS IN THE
      CAVES. THERE ARE GIANT BATS
      THAT WILL PICK YOU UP AND
      DROP YOU ELSEWHERE."
1160 VTAB 23: INPUT "HIT RETURN
      WHEN READY TO CONTINUE : ";A
      NS$
1170 HOME : VTAB 3: HTAB 12: PRINT
      "*** GRUE STEW ***": VTAB 7

```

```

1180 PRINT "THERE ARE BOTTOMLESS
      PITS. IF YOU FALL INTO ONE
      OF THESE YOU'LL NEVER GET O
      UT!"
1190 PRINT : PRINT "OF COURSE TH
      ERE IS THE GRUE HIMSELF,
      THOUGH NOT AN AGGRESSIVE CRE
      ATURE, HE WILL EAT YOU IF
      YOU COME TOO CLOSE."
1200 PRINT : PRINT "AND THERE AR
      E EARTHQUAKES THAT MOVE
      THINGS AROUND IN THE CAVES (
      BATS, PITS, THE GRUE, AND TH
      E EXIT!)."
1210 VTAB 23: INPUT "HIT RETURN
      WHEN READY TO CONTINUE : ";A
      NS$
1990 RETURN
2000 :
2001 REM *** SETUP
2002 :
2005 BELL$ = CHR$ (7)
2010 DIM RO$(20),TR(20,4)
2015 FOR I = 1 TO 20: READ RO$(I
      ): NEXT I
2020 DATA YOU ARE IN A SMALL
      ROOM WITH ROCKS AND DEBRIS
      SCATTERED EVERYWHERE.
2021 DATA DUCK YOUR HEAD IN
      HERE; AS LARGE ROCK STALA
      CTITES HANG FROM THE CEILING
      .
2022 DATA THE ROOM HERE SLOP
      ES DOWNWARD.
2023 DATA THIS ROOM IS VERY S
      MALL; BUT I THINK WE CAN MAK
      E IT THROUGH OK.
2024 DATA THIS IS A VERY LAR
      GE ROOM WITH A LARGE BOULDE
      R IN THE CENTER OF IT.
2025 DATA THIS IS THE CENTER OF
      A NARROW PASSAGE THAT CONN
      ECTS OTHER ROOMS.

```

2026 DATA THIS PASSAGE IS VERY  
LOW; BUT IF WE CRAWL WE  
CAN MAKE IT.

2027 DATA THIS IS A VERY DIRTY  
ROOM; IT HAS BEEN PARTIALL  
Y FILLED IN BY THE LAST EART  
H- QUAKE THAT HIT.

2028 DATA THIS ROOM IS ABOUT  
AVERAGE SIZE; BUT IS FILLED  
WITH A PUNGENT AROMA THAT I  
S VERYNAUSEATING.

2029 DATA YOU ARE IN A SMALL PA  
SSAGEWAY.

2030 DATA YOU ARE IN A SMALL PA  
SSAGEWAY.

2031 DATA YOU ARE IN A SMALL PA  
SSAGEWAY.

2032 DATA YOU ARE IN A SMALL PA  
SSAGEWAY.

2033 DATA YOU ARE IN A SMALL PA  
SSAGEWAY.

2034 DATA YOU ARE IN A SMALL PA  
SSAGEWAY.

2035 DATA A SMALL HOLE IN TH  
E CEILING LETS LIGHT FROM O  
UTSIDE THROUGH ... BUT YOU W  
OULD NEVER FIT THROUGH IT.

2036 DATA SOMEONE HAS LEFT A L  
IGHTED TORCH ON THE WALL AND  
IT ILLUMINATES YOUR PASSAGE

2037 DATA A RIVULET OF WATER SL  
OWLY TRICKLES FROM A HOLE IN  
THE WALL.

2038 DATA A SMALL HOLE TO YOUR  
LEFT ATTRACTS YOUR ATTENTION  
; BUT IT IS TOO SMALL TO BE  
OF ANY CONCERN.

2039 DATA YOU ARE IN A LOW  
DEPRESSION IN THE CENTE  
R OF A MEDIUM-SIZED ROOM.

2050 FOR I = 1 TO 20:F = 0

```

2055  FOR J = 1 TO 4: GOSUB 2955:
      F = F + TR(I,J): NEXT J: IF
      NOT F THEN 2055
2075  NEXT I
2100  YO =  INT ( RND (1) * 20) +
      1
2105  GU =  INT ( RND (1) * 20) +
      1
2106  IF GF THEN GU =  - 1
2110  EX =  INT ( RND (1) * 20) +
      1
2115  B1 =  INT ( RND (1) * 20) +
      1
2120  B2 =  INT ( RND (1) * 20) +
      1
2125  P1 =  INT ( RND (1) * 20) +
      1
2130  P2 =  INT ( RND (1) * 20) +
      1
2135  RETURN
2955  IF  INT ( RND (1) * 3) + 1 =
      2 OR TR(I,J) THEN RETURN
2960  RO =  INT ( RND (1) * 20) +
      1: IF RO = I THEN 2955
2961  DI =  INT ( RND (1) * 4) + 1
      : IF TR(RO,DI) THEN 2955
2965  TR(I,J) = RO:TR(RO,DI) = I
2990  RETURN
3000  :
3001  REM  *** PLAY
3002  :
3010  HOME : VTAB 3: HTAB 12: PRINT
      "*** GRUE STEW ***": VTAB 7
3015  PRINT : PRINT RO$(YO): FOR
      I = 1 TO 10:XX =  PEEK ( - 1
      6336): NEXT I
3020  FOR I = 1 TO 4:CO = TR(YO,I
      )
3021  IF CO = EX THEN  PRINT BELL
      $"EXIT NEARBY ...,"
3022  IF CO = GU THEN  PRINT BELL
      $"I SMELL THE GRUE !!!"

```



```

3023 IF CO = B1 OR CO = B2 THEN
      PRINT BELL$"FLAP... FLAP...
      FLAP ...,"
3024 IF CO = P1 OR CO = P2 THEN
      PRINT BELL$"I FEEL A DRAFT
      !!!"
3025 NEXT
3026 IF INT ( RND (1) * 15) = 4
      THEN PRINT BELL$"<<< EARTH
      QUAKE >>>": GOSUB 2105: GOTO
      3015
3030 PRINT : INPUT "MOVE OR SHOO
      T? ";ANS$: IF ANS$ = "S" THEN
      3500
3035 IF ANS$ < > "M" THEN PRINT
      "TYPE IN 'M' OR 'S'...": GOTO
      3030
3040 INPUT "WHICH WAY?";ANS$: FOR
      I = 1 TO 4: IF ANS$ < > MID$
      ("NESW",I,1) THEN NEXT : PRINT
      "ENTER 'N', 'E', 'W', OR 'S'
      ": GOTO 3030
3045 IF NOT TR(YO,I) THEN PRINT
      BELL$"YOU CANNOT GO THAT WAY
      ...": GOTO 3015
3050 PRINT "OK ...":YO = TR(YO,I
      )
3051 IF YO = EX THEN WL = 0: RETURN
3052 IF YO = GU THEN WL = 1: RETURN
3053 IF YO = P1 OR YO = P1 THEN
      WL = 2: RETURN
3054 IF YO < > B1 AND YO < > B
      2 THEN 3015
3055 PRINT "BATS HAVE YOU !!!": PRINT
      "THEY'RE LIFTING YOU UP !!!"
      : PRINT "OHHHHH, WHERE ARE W
      E NOW ???":YO = INT ( RND (
      1) * 20) + 1: GOTO 3015

```



```

3500 INPUT "SHOOT WHICH WAY? ";A
NS$: FOR I = 1 TO 4: IF ANS$
< > MID$ ("NESW",I,1) THEN
NEXT : PRINT "TYPE IN 'N',
'E', 'W', OR 'S'": GOTO 3030

3505 IF NOT TR(YO,I) THEN PRINT
BELL$"CLUNK!": PRINT "THE AR
ROW BOUNCED OFF THE WALL.": GOTO
3015

3510 IF TR(YO,I) = GU THEN PRINT
BELL$BELL$"OUCH !!!": PRINT
"YOU BAGGED A GRUE !!!": PRINT
"NOW TO FIND THE WAY OUT ...
":GF = 1:GU = - 1: GOTO 301
5

3520 PRINT BELL$"THE ARROW MISSE
D THE GRUE !!!": GOTO 3015

4000 :
4001 REM *** END
4002 :
4010 IF WL = 0 AND GF THEN PRINT
"YOU HAVE REACHED THE EXIT W
ITH YOUR": PRINT "GRUE !!! Y
OU WILL HAVE A FILLING SUPPE
R": PRINT "TONIGHT FOR SURE
!!!": RETURN

4015 IF WL = 0 THEN PRINT "YOU
HAVE REACHED THE EXIT WITHOU
T": PRINT "ANY GRUE !!! YOU
ARE SURE TO STARVE ..": RETURN

4020 IF WL = 1 THEN PRINT "YOU
BUMPED INTO THE GRUE !!!": PRINT
"HE ATE YOU BEFORE YOU COULD
MOVE !!!": RETURN

4025 IF WL = 2 THEN PRINT "YOU
FELL INTO A PIT !!!": PRINT
"YOU FELL A LOOOOOONG WAY ..
.": RETURN

```



A clever premise overshadows an interesting game. The IRSman serves as an excellent mathematical teaching device, while being entertaining and challenging. Everyone wants to beat the IRS, now here's your chance! To play, choose a number (we'll call it 'X'), and the digits 1 through X will appear. Each time you remove a number from the list, all of the factors of that number (which are still on the list) go to the IRSman. The object is to garner as much money as possible, while being as stingy as possible with the IRSman. If you play the number 12, 1-2-3-4-5-6-7-8-9-10-11-12 will appear on the screen. If you begin play by selecting 12, the IRSman will get 6, 2, 4, 3, and 1 ( $6 \times 2 = 12$ ,  $4 \times 3 = 12$ ,  $1 \times 12 = 12$ ) for a score of 16 to your 12. The board will now look like this: 11 10 9 8 7 5. As you will note, the only remaining number on the list which has a factor, is 10 (the remaining factor is 5). Remember, to remove a dollar amount (a number) from the list, there must be a factor to go to the IRSman. When you remove 10 from the above example, the score will be 22 ( $12 + 10$ ) for you, and 21 ( $16 + 5$ ) for the IRSman. But look . . . the list now reads: 11 9 8 7. None of these numbers has a factor left on the list, so they all revert to the IRSman. The final score would be 22 for you, and 56 ( $21 + 11 + 9 + 8 + 7$ ) for the IRSman. If you choose 6 before choosing 12, the IRSman scores for  $3 \times 2 = 6$ , (5 dollars) and  $1 \times 6$  (1 dollar) for a total of 6 dollars for him and 6 dollars for you. The digits may only be used once, so 6, 3, 2, and 1 are removed from the list. Now when you choose 12, the IRSman only gets 4 dollars ( $4 \times 3 = 12$ ). Now the score is 18 ( $6 + 12$ ) for you, and 10 ( $6 + 4$ ) for the IRSman. Note that 11 is stuck because the only factors of 11 are 11 and 1. Don't waste the universal factor (1) on just any number. It should be used first, to remove the highest prime number from the list. A prime number is one that is only divisible by itself and 1. Examples are 1, 2, 3, 5, 7, 11, 13, 17, 19, 23, etc. To circumvent the loss of 11, choose this number before choosing 6. There are

many ways to thwart the IRSman, but you must really try. Remember that all of the unused numbers (at the end of the game) are added to the score of the IRSman. The maximum score you can achieve when choosing 1 through 12, is 48.

```
10 REM *****
11 REM ***
12 REM *** IRSMAN ***
13 REM ***
14 REM *****
15 REM
16 REM
20 GOSUB 1000: REM INSTS
30 GOSUB 2000: REM SETUP
40 GOSUB 3000: REM PLAY!
50 GOSUB 4000: REM !END!
60 END
1000 :
1001 REM *** INSTS
1002 :
1010 TEXT : NORMAL : HOME
1020 VTAB 2: HTAB 13: PRINT "***
      IRSMAN ***"
1030 VTAB 5: PRINT "THIS IS THE
      GAME OF IRSMAN. TO WIN, YOU
      TRY TO ACCUMULATE MORE MONEY
      THAN YOUR NEMESIS, THE IRS
      MAN."
1033 PRINT
1035 PRINT "GIVE ME A NUMBER BET
      WEEN 1 AND 50. I WILL D
      ISPLAY A CONSECUTIVE NUMBER
      STRING STARTING AT 1, AN
      D CONTINUING ";
1036 PRINT "THROUGH TO THE NUMBE
      R YOU SELECTED. YOUWILL THE
      N CHOOSE HOW MUCH MONEY (WHI
      CH NUMBER) YOU WANT TO REMO
      VE FROM THE LIST."
1038 PRINT
1040 PRINT "BUT, AND HERE'S THE
      FUN PART, THE IRSMANGETS ALL
      OF THE REMAINING NUMBERS ON
      THE";
```

```

1042 PRINT "LIST THAT ARE FACTOR
      S OF THE NUMBER YOU CHOSE,
      THAT IS HOW THE IRSMAN GETS
      HIS MONEY. IF YOU CHOOSE 6,
      FOR EXAMPLE, ";
1043 PRINT "THE IRSMAN GETS ALL
      OF THE REMAINING FACTORS
      OF 6, (POTENTIALLY 1,2, AND
      3).";
1050 PRINT : INPUT "PRESS RETURN
      WHEN READY TO CONTINUE : ";
      ANS$
1055 TEXT : NORMAL : HOME
1060 VTAB 2: HTAB 13: PRINT "***
      IRSMAN ***"
1065 VTAB 5: PRINT "YOU CANNOT C
      HOOSE A NUMBER THAT HAS NO
      REMAINING FACTORS IN THE LIS
      T, BECAUSE YOU MUST ALWAYS
      PAY THE IRS."
1066 PRINT
1070 PRINT "WHEN YOU CAN NO LONG
      ER REMOVE ANY OF THEREMAININ
      G NUMBERS FROM THE LIST, THE
      IRSMAN CLAIMS ALL OF THE
      UNUSED MONEY (NUMBERS) FO
      R HIMSELF."
1080 VTAB 23: INPUT "PRESS RETUR
      N WHEN READY TO CONTINUE : "
      ;ANS$
1990 RETURN
2000 :
2001 REM *** SETUP
2002 :
2010 DIM LI(50): FOR I = 1 TO 50
      :LI(I) = I: NEXT
2020 VTAB 23: CALL - 958
2022 PRINT CHR$(7): INPUT "HOW
      MANY NUMBERS (1-50) IN THE
      LIST? ";ANS$

```



```

2025  ANS = VAL (ANS$): IF ANS <
      1 OR ANS > 50 OR ANS < > INT
      (ANS) THEN VTAB 22: CALL -
      958: PRINT : PRINT "<<< USE
      A NUMBER FROM 1 TO 50 >>>": FOR
      PA = 1 TO 2000: NEXT : GOTO
      2020
2030  NU = ANS
2990  RETURN
3000  :
3001  REM *** PLAY!
3002  :
3005  HOME : VTAB 3: HTAB 13: PRINT
      "*** IRSMAN ***": PRINT
3010  PRINT : PRINT "HERE IS THE
      LIST : "; FOR I = 1 TO NU: IF
      LI(I) THEN PRINT I" ";
3015  IF PEEK (36) > 35 THEN PRINT

3020  NEXT I
3021  IF NU = 1 THEN PRINT : PRINT
      : PRINT "OOOOPS, YOU CAN'T G
      ET ANYTHING...":TA = 1:LI(1)
      = 0: RETURN
3025  FOR I = 2 TO NU: IF NOT LI
      (I) THEN 3040
3030  -FOR J = 1 TO I: IF NOT LI(
      J) THEN 3035
3031  IF J = I THEN 3035
3033  IF LI(I) / J = INT (LI(I) /
      J) THEN 3050
3035  NEXT J
3040  NEXT I: RETURN
3050  PRINT : PRINT : PRINT "THE
      SCORE IS:   IRSMAN: "TA: PRINT
      "
      *YOU* : "YO
      U
3060  PRINT : INPUT "WHICH DO YOU
      WANT? ";ANS$
3065  ANS = VAL (ANS$): IF ANS <
      1 OR ANS > NU OR LI(ANS) = 0
      OR ANS < > INT (ANS) THEN
      PRINT : PRINT "THAT IS NOT
      AVAILABLE !": GOTO 3060

```

```

3070 SC = 0: IF AN = 1 THEN 3100
3075 FOR I = 1 TO AN: IF LI(I) =
    0 THEN 3090
3076 IF I = AN THEN 3090
3080 IF AN / I = INT (AN / I) THEN
    SC = SC + I
3090 NEXT I
3100 IF SC = 0 THEN PRINT : PRINT
    "YOU CAN'T HAVE IT. THAT LEA
    VES NOTHING FOR THE IRSMAN"
    : GOTO 3010
3105 LI(AN) = 0:YO = YO + AN:TA =
    TA + SC
3110 FOR I = 1 TO AN: IF LI(I) =
    0 THEN 3125
3115 IF I = AN THEN 3125
3120 IF AN / I = INT (AN / I) THEN
    LI(I) = 0
3125 NEXT I
3130 GOTO 3010
4000 :
4001 REM *** END
4002 :
4010 PRINT : PRINT : PRINT "***
    THE GAME IS OVER ***": PRINT

4015 FOR I = 1 TO NU: IF LI(I) THEN
    TA = TA + LI(I)
4016 NEXT
4020 PRINT "THE IRSMAN: "TA
4021 PRINT "          YOU: "YO
4022 PRINT "===== ": PRINT

4025 IF TA > YO THEN PRINT "THE
    IRSMAN IS THE WINNER !!!"
4030 IF TA < YO THEN PRINT "YOU
    HAVE BEATEN THE IRSMAN !!!"

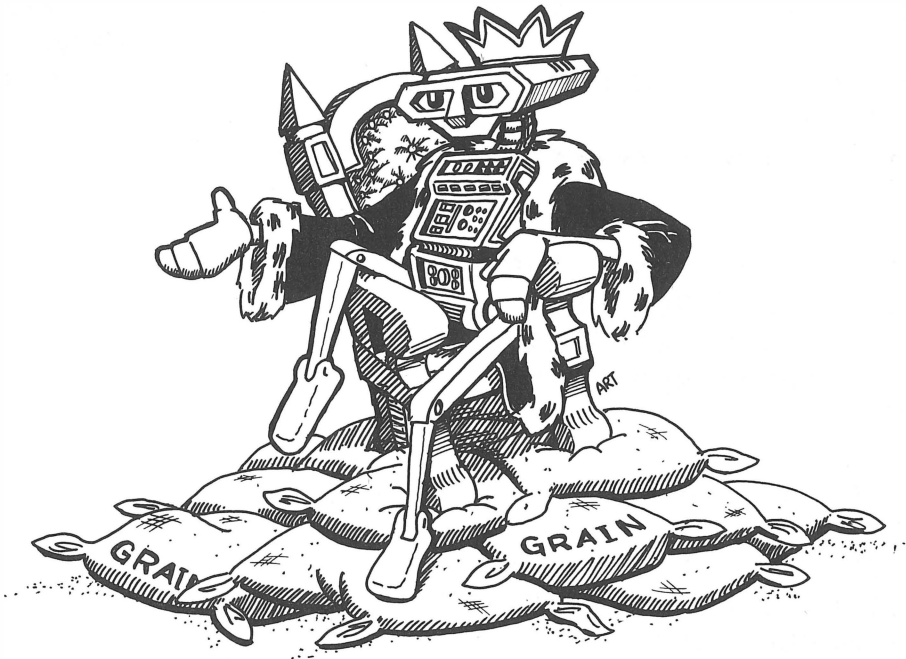
4035 IF TA = YO THEN PRINT "IT'S
    UNBELIEVABLE BUT ITS A TIE
    !!!"
4040 PRINT CHR$ (7); CHR$ (7); CHR$
    (7)
4990 RETURN

```









# Kingdom

This game is designed to test your leadership ability. You are given a ten year reign, during which time you try to guide your kingdom towards health and prosperity. There are certain conditions which are beyond your control (such as the bountiful nature of the harvest), but try to do the best job possible.

When the price of acreage is high, (25 or 26 bushels per acre), you may choose to become a land broker instead of a gentleman farmer. That is, you may sell all but one acre of land (you must keep 1), and hope the price of land drops the following year. If the price of land drops by 4 bushels (say, from 26 to 22), you have, in effect, made a 4 bushel per acre profit. When the price of land is low, (below 20 bushels for an acre), it is recommended that you buy as much land as possible, while retaining enough grain to feed your people and sow your fields. You can easily understand the program by manipulating the beginning values of the variables in line 2010 and changing some of the random number statements which are the unpredictable forces of nature.

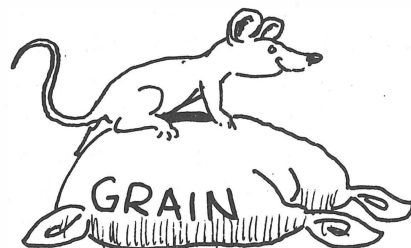
```

10 REM *****
11 REM *** ***
12 REM *** KINGDOM ***
13 REM *** ***
14 REM *****
15 REM
16 REM
20 GOSUB 1000: REM INSTS
30 GOSUB 2000: REM SETUP
40 GOSUB 3000: REM PLAY!
50 GOSUB 4000: REM !END!
60 END

1000 :
1001 REM *** INSTS
1002 :
1010 TEXT : NORMAL : HOME
1020 VTAB 3: HTAB 12: PRINT "***
      KINGDOM ***"
1030 VTAB 7: PRINT "THIS IS A SI
      MULATION OF THE COUNTRY OF
      SUMERIA. YOU ARE THE SOVERE
      IGN RULER, AND YOU WILL GOV
      ERN FOR 10 YEARS."
1040 PRINT : PRINT "THE DECISION
      S THAT YOU MAKE WILL AFFECT
      THE LIVES OF HUNDREDS OF PEO
      PLE. YOUR DICTATORIAL SKIL
      LS WILL BE RATED ONCE YOUR
      REIGN HAS ENDED."
1050 PRINT : PRINT "YOU WILL BE
      ASKED TO MAKE SEVERAL KEY
      DECISIONS EACH YEAR, WITH EA
      CH ONE BEING EXPLAINED TO YOU
      ."
1060 VTAB 23: INPUT "HIT RETURN
      WHEN READY TO CONTINUE : ";A
      NS$

1990 RETURN
2000 :
2001 REM *** SETUP
2002 :
2010 P = 95:S = 2800:H = 3000:E =
      H - S:Y = 3:A = H / Y:I = 5:
      D = 0:Z = 0:Q = 1

```



```

2020 DIM NU$(11): FOR J = 1 TO 1
      1: READ NU$(J): NEXT : DATA
      FIRST,SECOND,THIRD,FOURTH,F
      IFTH,SIXTH,SEVENTH,EIGHTH,NI
      NTH,TENTH,ELEVENTH
2990 RETURN
3000 :
3001 REM *** PLAY
3002 :
3005 HOME : VTAB 3: HTAB 12: PRINT
      "*** KINGDOM ***": VTAB 7
3010 Z = Z + 1: PRINT : PRINT "HA
      MURABI, I BEG TO REPORT TO Y
      OU: ": PRINT : PRINT "IN THE
      "NU$(Z)" YEAR, "D" PEOPLE "
      : PRINT "STARVED; "I" CAME T
      O THE CITY."
3280 P = P + I: IF Q = 0 THEN P =
      INT (P / 2): PRINT : PRINT
      "A HORRIBLE PLAGUE STRUCK !!
      ! HALF OF YOUR PEOPLE PER
      ISHED...."
3285 PRINT
3290 PRINT "THE POPULATION IS "P
      ", THE CITY OWNS ": PRINT A
      " ACRES. YOU HARVESTED "Y"
      BUSHEL": PRINT "PER ACRE,
      RATS ATE "E" BUSHEL": PRINT
      "YOU HAVE "S" BUSHEL IN RES
      ERVE."
3300 IF Z = 11 THEN RETURN
3400 C = INT ( RND (1) * 10):Y =
      C + 17
3410 PRINT : PRINT "LAND IS TRAD
      ING AT "Y" BUSHEL PER ACRE.
      ": PRINT "HOW MANY ACRES DO
      YOU WISH TO BUY : ": INPUT Q

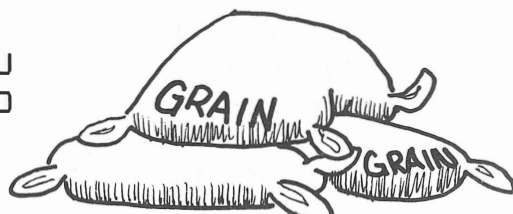
3440 IF Q < 0 THEN PRINT "HAMUR
      ABI, YOU CANNOT DO THAT.": PRINT
      "IF YOU WISH TO SELL LAND, "
      : PRINT "FIRST BUY 0 ACRES."
      : GOTO 3410

```

```

3450 IF Y * Q > S THEN PRINT "H
      AMURABI, THINK AGAIN! YOU O
      NLY HAVE": PRINT S" BUSHEL
      OF GRAIN.": GOTO 3410
3455 IF Q > 0 THEN A = A + Q:S =
      S - Y * Q:C = 0: GOTO 3500
3460 INPUT "HOW MANY ACRES DO YO
      U WISH TO SELL?";Q
3465 IF Q < 0 THEN PRINT "HAMUR
      ABI, I CANNOT DO THAT. IF Y
      OU": PRINT "DON'T WANT TO SE
      LL ANYTHING, THEN": PRINT "S
      ELL 0 ACRES.": GOTO 3460
3470 IF (Q > A) THEN PRINT "HAM
      URABI, YOU ONLY OWN "A" ACRE
      S....": GOTO 3460
3480 A = A - Q:S = S + Y * Q:C =
      0
3500 PRINT : PRINT "OF THE "S" B
      USHELs REMAINING, HOW MANY":
      PRINT "DO YOU WISH TO FEED
      YOUR PEOPLE:": INPUT Q
3505 IF Q < 1 THEN PRINT "HAMUR
      ABI, THE PEOPLE WILL STARVE
      !!!": PRINT "YOU MUST FEED T
      HEM SOMETHING.": GOTO 3500
3510 IF Q > S THEN PRINT "HAMUR
      ABI, YOU ONLY OWN "S" BUSHEL
      S ...": GOTO 3500
3520 S = S - Q:C = 1
3530 PRINT : PRINT "OF THE "A" A
      CRES YOU NOW OWN, HOW": INPUT
      "MANY DO YOU WISH TO PLANT W
      ITH SEED? ";D
3535 IF D < 1 THEN PRINT "HAMUR
      ABI, YOU MUST PLANT SOMETHIN
      G SO": PRINT "THAT THERE WIL
      L BE FOOD FOR NEXT YEAR... "
      : GOTO 3530
3540 IF (D > A) THEN PRINT "YOU
      ONLY HAVE "A" ACRES.": GOTO
      3530

```



```

3545 IF D / 2 > S THEN PRINT "H
      AMURABI, THAT IS TOO MUCH TO
      PLANT...": GOTO 3530
3550 IF D > 10 * P THEN PRINT "
      YOU CAN ONLY FORCE ONE PERSON
      TO ": PRINT "WORK TEN ACRES
      OF LAND.": PRINT "YOUR POPU-
      LATION OF "P" ISN'T BIG ENO-
      UGH.": GOTO 3530
3555 S = S - INT (D / 2):C = INT
      ( RND (1) * 5) + 1
3600 Y = C:H = D * Y:E = 0:C = INT
      ( RND (1) * 5) + 1: IF INT
      (C / 2) * 2 = C THEN E = INT
      (S / C)
3610 S = S - E + H:C = INT ( RND
      (1) * 5) + 1:I = INT (C * (
      20 * A + S) / P / 100 + 1):C
      = INT (Q / 20):Q = INT (1
      0 * (2 * RND (1) - .3)): IF
      P < C THEN D = 0: GOTO 3010
3615 D = P - C: IF D > .50 * P THEN
      3630
3620 P1 = ((Z - 1) * P1 + D * 100
      / P) / Z:P = C:D1 = D1 + D:
      GOTO 3010
3630 PRINT : PRINT "YOU STARVED
      "D" PEOPLE IN ONE YEAR !": PRINT
      "YOU HAVE DONE SUCH A MISERA-
      BLE JOB": PRINT "THAT YOU HA-
      VE BEEN OVERTHROWN": PRINT "
      AND REMOVED FROM OFFICE !!!"
      :WL = 1: RETURN
4000 :
4001 REM *** END
4002 :
4005 IF WL THEN RETURN
4010 PRINT : PRINT : PRINT "IN Y
      OUR 10 YEARS OF RULE, "P1"%
      ": PRINT "OF THE POPULATION
      STARVED PER YEAR, ON": PRINT
      "THE AVERAGE.": PRINT "A TOT-
      AL OF "D1" PEOPLE DIED.":L =
      A / P

```

```

4015 PRINT
4020 PRINT "YOU STARTED WITH 10
      ACRES PER PERSON, AND ENDE
      D WITH "L" ACRES PER PER
      SON !!"
4030 IF P1 > 33 OR L < 7 THEN PRINT
      "YOU ARE A DISGRACE!!! THE
      PEOPLE HAVE EXILED YOU TO A
      REMOTE ISLAND.": RETURN
4035 IF P1 > 10 OR L < 9 THEN PRINT
      "YOU RULE LIKE THE AYATOLLAH
      ! MOST OF YOUR SUBJECTS W
      OULD DANCE AT YOUR FUN
      ERAL!": RETURN
4040 IF P1 > 3 OR L < 10 THEN PRINT
      "YOU COULD HAVE DONE BETTER,
      " INT (P * .8 * RND (1))"
      PEOPLE": PRINT "WOULD LOVE T
      O SEE YOU ASSASSINATED!!!": RETURN
4045 PRINT : PRINT "A GREAT JOB!
      !! YOU CAN RULE MY COUNTRY
      ANY TIME YOU WANT TO !!!": RETURN

```

In this game you try to alphabetize a scrambled list of letters. One square is left blank so that you may move a letter into it. The computer will scramble the completed version approximately 150 times. You must unscramble the letter matrix in as few tries as possible. Remember, practice makes perfect! Before we look at the graphics, it seems appropriate to mention that proper graphics alignment is not a matter of trial and error, but entails working with graph paper and drawing the figure which is to be outputted to the screen. Then, after figuring which coordinates will be filled in with which color, the writing of the program can take place. The nice pictures that you see on the screen are more the result of painstaking work than the result of brilliance. Getting back to Magic Squares, let's look at some of the graphics. Lines 2025-2028 draw the gameboard. Lines 2221-2235 draw the letters onto the game-board. Check this out by typing: 2225 STOP, then run. The first four letters (A,B,C, and D) are drawn. Type: CONT to continue the run. A matrix with two F's will be displayed. To undo any change, type: LOAD (program name). When this is done, the original program (without any changes) will be loaded.

```

10  REM  *****
11  REM  ***                               ***
12  REM  *** MAGIC SQUARES ***
13  REM  ***                               ***
14  REM  *****
15  REM
16  REM
20  GOSUB 1000: REM  INSTS
30  GOSUB 2000: REM  SETUP
40  GOSUB 3000: REM  PLAY!
50  GOSUB 4000: REM  !END!
60  END
1000
1001  REM  *** INSTS
1002
1010  TEXT : NORMAL : HOME
1020  VTAB 3: HTAB 9: PRINT "***
      MAGIC SQUARES ***"
1030  VTAB 7: PRINT "IN THIS GAME
      , YOU HAVE A 4 BY 4 GAME-
      BOARD. THE BOARD CONTAINS T
      HE LETTERS A-O."
1035  PRINT
1040  PRINT "THE OBJECT IS TO HOR
      IZONTALLY          ALPHABET
      IZE THE SCRAMBLED GAME-BOARD
      YOU CAN MOVE A PIECE SID
      EWAYS OR UP AND DOWN, AS LON
      G AS THE EMPTY SQUARE IS
      NEXT TO IT. "
1045  PRINT
1050  PRINT "YOUR PROGRESS WILL B
      E MONITORED, AND YOUWILL BE
      TOLD HOW YOU ARE DOING. "
1055  VTAB 23: INPUT "PRESS RETUR
      N WHEN READY TO CONTINUE : "
      ;ANS$
1990  RETURN
2000
2001  REM  *** SETUP
2002
2005  DEF FN C(X) = (X - 1) * 8 +
      2

```



```

2010 DIM B(4,4): FOR I = 1 TO 4:
    FOR J = 1 TO 4: K = K + 1: B(
J,I) = K: NEXT J,I
2020 DIM DIR(4,2): FOR I = 1 TO
4: READ DIR(I,1),DIR(I,2): NEXT
: DATA 1,0,0,1,-1,0,0,-1
2025 GR : COLOR= 12
2026 FOR I = 1 TO 32: HLIN 0,32 AT
I: NEXT
2027 COLOR= 15
2028 FOR I = 0 TO 32 STEP 8: HLIN
0,32 AT I: VLIN 0,32 AT I: NEXT

2029 CO = 15: GOSUB 2100
2030 VTAB 23: PRINT "<< I'M NOW
SCRAMBLING THE GAME BOARD >>
"
2040 SX = 4:SY = 4:SC = INT ( RND
(1) * 50) + 100: FOR K = 1 TO
SC
2050 D = INT ( RND (1) * 4) + 1:
PX = SX + DI(D,1):PY = SY +
DI(D,2): IF PX < 1 OR PX > 4
OR PY < 1 OR PY > 4 THEN 20
50
2060 B(SX,SY) = B(PX,PY)
2061 J = SX:I = SY:CO = 14: GOSUB
2200
2062 J = PX:I = PY:CO = 12: GOSUB
2200
2065 B(PX,PY) = 0: SX = PX:SY = PY

2070 NEXT K: RETURN
2100 REM *** DRAW BOARD
2110 FOR I = 1 TO 4: FOR J = 1 TO
4: CO = 14: GOSUB 2200: NEXT
J,I: RETURN
2200 REM *** DRAW LETTER
2205 X = FN C(J):Y = FN C(I): COLOR=
CO

```

```

2210  ON B(J,I) + 1 GOSUB 2220,22
      21,2222,2223,2224,2225,2226,
      2227,2228,2229,2230,2231,223
      2,2233,2234,2235: RETURN
2220  RETURN
2221  PLOT X + 2,Y: PLOT X + 1,Y +
      1: PLOT X + 3,Y + 1: VLIN Y +
      2,Y + 4 AT X: VLIN Y + 2,Y +
      4 AT X + 4: HLIN X,X + 4 AT
      Y + 3: RETURN
2222  VLIN Y,Y + 4 AT X: HLIN X,X
      + 3 AT Y: HLIN X,X + 3 AT Y
      + 2: HLIN X,X + 3 AT Y + 4:
      PLOT X + 4,Y + 1: PLOT X +
      4,Y + 3: RETURN
2223  HLIN X,X + 4 AT Y: HLIN X,X
      + 4 AT Y + 4: VLIN Y,Y + 4 AT
      X: RETURN
2224  HLIN X,X + 3 AT Y: HLIN X,X
      + 3 AT Y + 4: VLIN Y,Y + 4 AT
      X: VLIN Y + 1,Y + 3 AT X + 4
      : RETURN
2225  HLIN X,X + 4 AT Y: HLIN X,X
      + 3 AT Y + 2: HLIN X,X + 4 AT
      Y + 4: VLIN Y,Y + 4 AT X: RETURN

2226  HLIN X,X + 4 AT Y: HLIN X,X
      + 3 AT Y + 2: VLIN Y,Y + 4 AT
      X: RETURN
2227  HLIN X,X + 3 AT Y: HLIN X,X
      + 4 AT Y + 4: VLIN Y,Y + 4 AT
      X: VLIN Y + 2,Y + 4 AT X + 4
      : PLOT X + 3,Y + 2: RETURN
2228  HLIN X,X + 4 AT Y + 2: VLIN
      Y,Y + 4 AT X: VLIN Y,Y + 4 AT
      X + 4: RETURN
2229  HLIN X,X + 4 AT Y: HLIN X,X
      + 4 AT Y + 4: VLIN Y,Y + 4 AT
      X + 2: RETURN
2230  HLIN X,X + 4 AT Y: HLIN X +
      1,X + 2 AT Y + 4: PLOT X,Y +
      3: VLIN Y,Y + 4 AT X + 2: RETURN

```

```

2231  VLIN Y,Y + 4 AT X: PLOT X +
      1,Y + 2: PLOT X + 2,Y + 1: PLOT
      X + 2,Y + 3: PLOT X + 3,Y: PLOT
      X + 3,Y + 4: RETURN
2232  HLIN X,X + 4 AT Y + 4: VLIN
      Y,Y + 4 AT X: RETURN
2233  VLIN Y,Y + 4 AT X: VLIN Y,Y
      + 4 AT X + 4: PLOT X + 1,Y +
      1: PLOT X + 3,Y + 1: PLOT X +
      2,Y + 2: RETURN
2234  VLIN Y,Y + 4 AT X: VLIN Y,Y
      + 4 AT X + 4: PLOT X + 1,Y +
      1: PLOT X + 2,Y + 2: PLOT X +
      3,Y + 3: RETURN
2235  VLIN Y,Y + 4 AT X: VLIN Y,Y
      + 4 AT X + 4: HLIN X,X + 4 AT
      Y: HLIN X,X + 4 AT Y + 4: RETURN

2990  RETURN
3000  :
3001  REM   *** PLAY!
3002  :
3010  HOME : PRINT "MOVE WHICH PI
      ECE : " : GET AN$: PRINT AN
      S$
3015  IF AN$ < "A" OR AN$ > "Z"
      THEN PRINT CHR$ (7): GOTO
      3010
3020  FOR K = 1 TO 4: PX = SX + DI
      R(K,1): PY = SY + DIR(K,2)
3025  IF PX < 1 OR PX > 4 OR PY <
      1 OR PY > 4 THEN 3040
3030  IF B(PX,PY) = ASC (AN$) -
      64 THEN MO = MO + 1: GOTO 31
      00
3040  NEXT K: PRINT CHR$ (7): GOTO
      3010
3100  B(SX,SY) = B(PX,PY): J = SX:I
      = SY: CO = 14: GOSUB 2200
3110  J = PX:I = PY: CO = 12: GOSUB
      2200: B(PX,PY) = 0: SX = PX: SY
      = PY

```



```

3200 K = 0: FOR I = 1 TO 4: FOR J
    = 1 TO 4: K = K + 1: IF K =
    16 THEN K = 0
3210 IF B(J,I) = K THEN NEXT J,
    I: RETURN
3220 GOTO 3010
3990 RETURN
4000 :
4001 REM *** END
4002 :
4010 HOME : PRINT "YOU SOLVED IT
    !": PRINT : PRINT "IT WAS S
    CRAMBLED "SC" TIMES, ": PRINT
    "AND YOU SOLVED IT IN "MO" M
    OVES. "
4020 INPUT "DO YOU WISH TO PLAY
    AGAIN? ";ANS$: IF LEFT$ (AN
    S$,1) = "Y" THEN RUN
4990 RETURN

```

This game is a spinoff from a popular game show. The object is to eliminate as many numbers as you can from the list before you get stymied. To begin, you are given a list of numbers ranging from 1 through 9. A pair of dice are rolled. The total (2-12) must be subtracted from the list. Most of the numbers can be removed from the list in a multitude of ways. If the first number rolled is a nine, there are eight possible ways to total exactly nine. They are:

1,2,6 1,3,5 1,8 2,3,4 2,7 3,6 4,5 and 9 by itself.

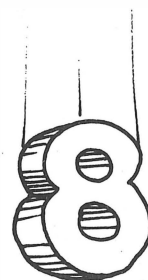
According to the rules, you may remove any of these combinations as long as the total is nine. There are many different strategies, but you can develop your own. Let's look at the graphics. Lines 2131 through 2139 draw the list of nine numbers. To verify this, type: 2136 and then return. Now when you run the program you will get 1 2 3 4 5 (error message). Line 2210 gives each die a random result between 1 and 6. To help you understand how one of the numbers is set to blinking, look at lines 2200-2336, and experiment with any of these lines.

```

10 REM *****
11 REM *** ***
12 REM *** NUMBERS AWAY ***
13 REM *** ***
14 REM *****
15 REM
16 REM
20 GOSUB 1000: REM INSTS
30 GOSUB 2000: REM SETUP
40 GOSUB 3000: REM PLAY!
50 GOSUB 4000: REM !END!
60 END

1000 :
1001 REM *** INSTS
1002 :
1010 TEXT : HOME : NORMAL
1020 VTAB 3: HTAB 10: PRINT "***
    NUMBERS AWAY ***"
1030 VTAB 7: PRINT "IN THIS GAME
    , YOU WILL BE PRESENTED WITH
    A LIST OF NUMERS BETWEEN 1 A
    ND 9."
1035 PRINT : PRINT "A PAIR OF DI
    CE WILL BE ROLLED, AND THE
    TOTAL WILL BE NOTED. YOU MU
    ST REMOVE, FROM THE LIST, A
    COMBINATION OF NUMBERS WHOS
    E TOTAL MATCHES THE NUMBER O
    N THE DICE."
1040 PRINT : PRINT "FOR EXAMPLE,
    IF A SEVEN WAS ROLLED, YOU
    COULD REMOVE FROM THE LIST (
    1,2,4), (1,6), (2,5), (3
    ,4) OR JUST PLAIN (7). "
1045 VTAB 23: INPUT "HIT RETURN
    WHEN READY TO CONTINUE : ";A
    NS$
1050 HOME : VTAB 3: HTAB 10: PRINT
    "*** NUMBERS AWAY ***": VTAB
    7

```



```

1055  PRINT "TO MOVE IN THE LIST,
      USE THE FORWARD ANDBACKWARD
      ARROWS. THE NUMBER YOU ARE
      AT WILL BLINK. TO SELECT A
      NUMBER, PUSH THEReturn KEY."

1060  PRINT : PRINT "WHEN YOU SEL
      ECT ENOUGH NUMBERS TO REACH
      THE TOTAL ON THE DICE, THE C
      OMPUTER WILLROLL THE DICE FO
      R YOUR NEXT TRY."

1065  PRINT : PRINT "IF YOUR TOTA
      L GOES OVER THE NUMBER, THE
      LIST WILL BE RESTORED, AND Y
      OU WILL HAVETO TRY AGAIN. TO
      GIVE UP, PRESS THE ESC KEY.
      "

1070  VTab 23: INPUT "HIT RETURN
      WHEN READY TO CONTINUE : ";A
      NS$

1990  RETURN
2000
2001  REM *** SETUP
2002
2010  DIM LI(9),L2(9): FOR I = 1 TO
      9:LI(I) = I: NEXT
2020  GR : HOME
2030  FOR I = 1 TO 9:NU = I: GOSUB
      2100: NEXT
2095  RETURN
2100  COLOR= NU: GOTO 2120
2110  COLOR= 0
2120  ON NU GOTO 2131,2132,2133,2
      134,2135,2136,2137,2138,2139

2131  HLin 2,3 AT 5: VLin 5,9 AT
      3: HLin 2,4 AT 9: RETURN
2132  HLin 6,8 AT 5: VLin 5,7 AT
      8: HLin 6,8 AT 7: VLin 7,9 AT
      6: HLin 6,8 AT 9: RETURN
2133  HLin 10,12 AT 5: HLin 10,12
      AT 7: HLin 10,12 AT 9: VLin
      5,9 AT 12: RETURN

```

```

2134  VLIN 5,7 AT 14: VLIN 5,9 AT
      16: HLIN 14,16 AT 7: RETURN

2135  HLIN 18,20 AT 5: HLIN 18,20
      AT 7: HLIN 18,20 AT 9: VLIN
      5,7 AT 18: VLIN 7,9 AT 20: RETURN

2136  HLIN 22,24 AT 5: HLIN 22,24
      AT 7: HLIN 22,24 AT 9: VLIN
      5,9 AT 22: VLIN 7,9 AT 24: RETURN

2137  HLIN 26,28 AT 5: VLIN 5,9 AT
      28: RETURN

2138  HLIN 30,32 AT 5: HLIN 30,32
      AT 7: HLIN 30,32 AT 9: VLIN
      5,9 AT 30: VLIN 5,9 AT 32: RETURN

2139  HLIN 34,36 AT 5: HLIN 34,36
      AT 7: VLIN 5,7 AT 34: VLIN
      5,9 AT 36: RETURN
2200  FOR J = 1 TO INT ( RND (1)
      * 5) + 5
2210  D1 = INT ( RND (1) * 6) + 1
      :D2 = INT ( RND (1) * 6) +
      1
2220  GOSUB 2300: GOSUB 2310
2230  NEXT
2300  DD = D1:DX = 10: GOTO 2320
2310  DD = D2:DX = 24: GOTO 2320
2320  COLOR= INT ( RND (1) * 14)
      + 1
2325  FOR I = 30 TO 36: HLIN DX,D
      X + 6 AT I: NEXT
2330  COLOR= 15: ON DD GOTO 2331,
      2332,2333,2334,2335,2336
2331  PLOT DX + 3,33: GOTO 2340
2332  PLOT DX + 1,31: PLOT DX + 5
      ,35: GOTO 2340
2333  PLOT DX + 1,31: PLOT DX + 3
      ,33: PLOT DX + 5,35: GOTO 23
      40
2334  PLOT DX + 1,31: PLOT DX + 5
      ,31: PLOT DX + 1,35: PLOT DX
      + 5,35: GOTO 2340

```





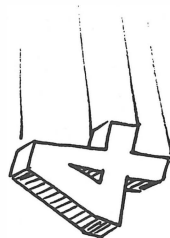
```

2335 PLOT DX + 1,31: PLOT DX + 5
    ,31: PLOT DX + 1,35: PLOT DX
    + 5,35: PLOT DX + 3,33: GOTO
2340
2336 PLOT DX + 1,31: PLOT DX + 5
    ,31: PLOT DX + 1,35: PLOT DX
    + 5,35: PLOT DX + 1,33: PLOT
    DX + 5,33: GOTO 2340
2340 FOR I = 1 TO 5:XX = PEEK (
    - 16336): NEXT : RETURN
3000 :
3001 REM *** PLAY
3002 :
3010 FOR I = 1 TO 9: IF NOT LI(
    I) THEN NEXT : HOME : PRINT
    CHR$ (7) CHR$ (7) CHR$ (7)"
    YOU GOT THEM ALL !!!": FOR I
    = 1 TO 1000: NEXT I: RETURN

3012 GOSUB 2200: REM ROLL DICE
3015 FOR I = 1 TO 9:L2(I) = 0: NEXT

3020 TT = D1 + D2:ST = 0
3025 HOME : PRINT "<<< YOU MUST
    GET A TOTAL OF "TT" >>>"
3030 FOR I = 1 TO 9: IF NOT LI(
    I) THEN NEXT : HOME : PRINT
    CHR$ (7)"THERE IS NOTHING L
    EFT, AND": PRINT "YOU CANNOT
    REACH THE TOTAL": FOR I = 1
    TO 1000: NEXT I: RETURN
3035 NP = 0: GOSUB 3100
3040 NU = LI(NP): GOSUB 2110: FOR
    I = 1 TO 50: NEXT I: GOSUB 2
    100
3050 IF PEEK ( - 16384) < 128 THEN
3040
3055 KEY = PEEK ( - 16384): POKE
    - 16368,0
3060 IF KEY = 149 THEN GOSUB 31
    00: GOTO 3040
3061 IF KEY = 136 THEN GOSUB 32
    00: GOTO 3040
3065 IF KEY = 155 THEN RETURN

```



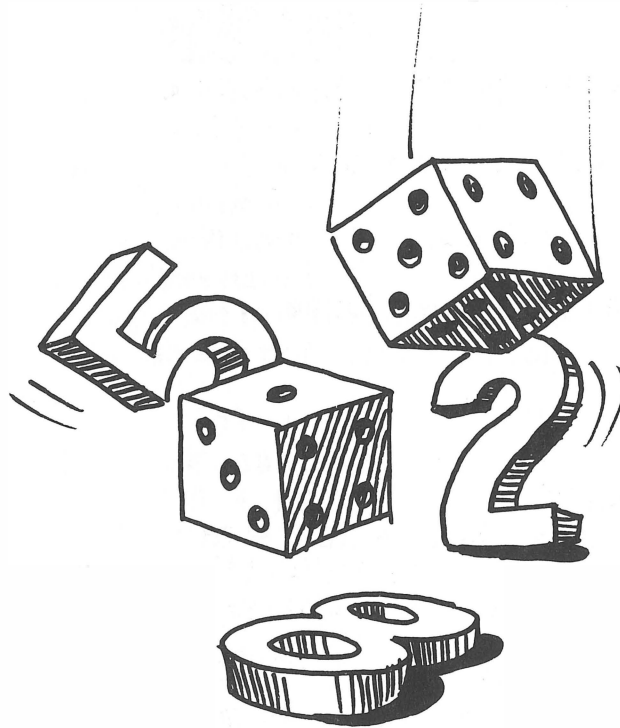
```

3070 IF KEY < > 141 THEN 3040
3072 NU = NP: GOSUB 2110
3075 ST = ST + NP: LI(NP) = 0: L2(N
P) = 1: IF ST < TT THEN PRINT
"YOU'VE GOT "ST", YOU NEED "
TT - ST"...": GOTO 3030
3080 IF ST = TT THEN GT = GT + T
T: HOME : PRINT CHR$ (7) CHR$
(7) CHR$ (7)"YOU GOT THAT ON
E !!!": FOR I = 1 TO 500: NEXT
I: GOTO 3010
3085 FOR I = 1 TO 9: IF L2(I) THEN
NU = I: GOSUB 2100: LI(I) = I

3090 NEXT :ST = 0: PRINT : PRINT
"OOOPS, TRY AGAIN! YOU NEED
"TT"...": CHR$ (7): GOTO 303
0
3100 NP = NP + 1: IF NP > 9 THEN
NP = 1
3110 IF LI(NP) = 0 THEN 3100
3120 RETURN
3200 NP = NP - 1: IF NP < 1 THEN
NP = 9
3210 IF LI(NP) = 0 THEN 3200
3220 RETURN
4000 :
4001 REM *** END
4002 :
4010 TEXT : HOME
4020 VTAB 3: HTAB 10: PRINT "***
NUMBERS AWAY ***": VTAB 15
4030 PRINT "YOU GOT "GT" OUT OF
A POSSIBLE 45.
4032 PRINT "THAT IS ";
4035 ON INT (GT / 5) GOTO 4040,
4041,4042,4043,4044,4045,404
6,4047,4048,4049
4040 PRINT "THE ABSOLUTE WORST !
!!": RETURN
4041 PRINT "EXTREMELY POOR !!!":
RETURN
4042 PRINT "TERRIBLE !!!": RETURN

```

```
4043 PRINT "VERY BAD !!!": RETURN
4044 PRINT "JUST SO-SO !!!": RETURN
4045 PRINT "FAIR ...": RETURN
4046 PRINT "PRETTY GOOD ...": RETURN
4047 PRINT "GREAT !!!": RETURN
4048 PRINT "FANTASTIC !!!!!!": RETURN
4049 PRINT "PERFECT !!!!!!": RETURN
```







This game can be frustrating, challenging, and exciting all at the same time. You are given a list of integers which you must unscramble using a reversing technique. The list is established in lines 2010 by setting up an array table using a FOR NEXT loop. LI (I) becomes LI (1), LI (2) etc. Line 2020 assigns the random order to the list. Line 3050 does the reversing of the numbers that you select as ANS in 3040. These lines have several commands grouped together for speed of operation. If you want to dissect them it is best to rewrite the line as separate statements. You can also print out the variables:

```

3050 MDL = INT ((9 - ANS) / 2)
3051 PRINT "MDL IN LINE 3051 =";MDL
3052 FOR I = ANS TO ANS + MDL
3053 PRINT "I =";I;" ANS + MDL ="; ANS+MDL
3054 T = LI(I)
3055 PRINT "T =";T
3056 LI(I) = LI(9 + ANS - I)
3057 PRINT "LI (I) =";LI(I)
3058 LI(9 + ANS - I) = T
3059 PRINT "T =";T:NEXT

```

This elaboration of line 3050 will let you watch the program pass the variables using 'T' as a temporary storage location. Practically all the action takes place in this one line. When you run this revised program, print statements will tell you what happens each time you make a reversal.

The purpose of combining statements on a single line is speed of execution. When you are developing programs you should have every statement on a separate line. When the program is running correctly you should make two versions. Make a long version with lots of REMS and descriptive variable names, and a short version with combined lines. This will help you modify the program later on. Your own program that you knew my heart a few weeks ago can become a complete mystery if you don't spend enough time on REMS and organization.

```

10 REM *****
11 REM ***          ***
12 REM ***    REVERSER    ***
13 REM ***          ***
14 REM *****
15 REM
16 REM
20 GOSUB 1000: REM  INSTS
30 GOSUB 2000: REM  SETUP
40 GOSUB 3000: REM  PLAY!
50 GOSUB 4000: REM  !END!
60 END

1000 :
1001 REM  *** INSTS
1002 :
1010 TEXT : HOME : NORMAL
1020 VTAB 2: HTAB 12: PRINT "***
      REVERSER ***"
1030 VTAB 5: PRINT "IN THIS GAME
      , YOU ARE GIVEN A LIST OF
      NUMBERS FROM 0 TO 9.  THE LI
      ST WILL NOT BE IN SEQUENCE.
      IT IS YOUR JOB TO SORT THE
      LIST INTO ASCENDING ORDER."
1040 PRINT : PRINT "YOU ARRANGE
      THE LIST BY REVERSING THE
      ORDER OF IT.  YOU INPUT THE
      STARTING    COLUMN THAT IS T
      O BE REVERSED, AND THAT COLU
      MN, ALL THE WAY THROUGH TO C
      OLMN    NINE, WILL BE REVERS
      ED."
1050 VTAB 23: INPUT "HIT RETURN
      WHEN READY TO CONTINUE :";AN
      S$

```



```

1060 HOME : VTAB 2: HTAB 12: PRINT
    "*** REVERSER ***": VTAB 5
1070 PRINT "IF YOU HAD THIS LIST
    ": PRINT : PRINT "POSITIONS
    : 0 1 2 3 4 5 6 7 8 9": PRINT
    "
    -----
    ---": PRINT "      LIST: 0 1
    9 8 7 6 5 2 3 4": PRINT
1080 PRINT "AND YOU REVERSED AT
    POSITION 7, IT WOULD LOOK LIK
    E THIS:"
1090 PRINT : PRINT "POSITIONS: 0
    1 2 3 4 5 6 7 8 9": PRINT "
    -----
    --": PRINT "      LIST: 0 1 9
    8 7 6 5 4 3 2": PRINT
1100 VTAB 23: INPUT "HIT RETURN
    WHEN READY TO CONTINUE : ";A
    NS$
1110 HOME : VTAB 3: HTAB 12: PRINT
    "*** REVERSER ***": VTAB 7
1120 PRINT "A FINAL REVERSAL AT
    POSITION 2 WOULD COMPLETE
    THE LIST AS THIS:"
1130 PRINT : PRINT "POSITIONS: 0
    1 2 3 4 5 6 7 8 9": PRINT "
    -----
    --": PRINT "      LIST: 0 1 2
    3 4 5 6 7 8 9": PRINT
1140 PRINT : PRINT "YOU WIN WHEN
    THE LIST IS SORTED IN
    ASCENDING ORDER AS IS SHOWN
    ABOVE.": PRINT : PRINT "GOOD
    LUCK !!!"
1150 VTAB 23: INPUT "HIT RETURN
    WHEN READY TO CONTINUE : ";A
    NS$
1990 RETURN
2000 :
2001 REM *** SETUP
2002 :
2010 DIM LI(9): FOR I = 0 TO 9:L
    I(I) = I: NEXT

```



```

2020  FOR I = 0 TO 9:X = INT ( RND
      (1) * 10):T = LI(I):LI(I) =
      LI(X):LI(X) = T: NEXT
2990  RETURN
3000  :
3001  REM *** PLAY
3002  :
3010  HOME : VTAB 3: HTAB 12: PRINT
      "*** REVERSER ***": VTAB 7
3020  PRINT : PRINT "POSITIONS: 0
      1 2 3 4 5 6 7 8 9": PRINT "
      -----
      --": PRINT "      LIST: ";
3030  FOR I = 0 TO 9: PRINT LI(I)
      " ";: NEXT : PRINT
3035  FOR I = 0 TO 9: IF LI(I) =
      I THEN NEXT : RETURN
3040  PRINT : INPUT "REVERSE AT W
      HIGH POSITION (0-9) ?";ANS$:
      ANS = VAL (ANS$): IF ANS <
      0 OR ANS > 9 OR ANS < > INT
      (ANS) THEN PRINT : PRINT "T
      YPE A NUMBER BETWEEN 0-9):GO
      TO 3040
3050  MDL = INT ((9 - ANS) / 2): FOR
      I = ANS TO ANS + MDL:T = LI(
      I):LI(I) = LI(9 + AN - I):LI
      (9 + AN - I) = T: NEXT
3060  MOVE = MOVE + 1: GOTO 3020
4000  :
4001  REM *** END
4002  :
4010  PRINT : PRINT "YOU DID IT !
      ": PRINT : PRINT "IT ONLY TO
      OK YOU "MO" MOVES"
4020  PRINT
4030  IF MO < 15 THEN PRINT "THA
      T'S SUPER!!!": RETURN
4040  PRINT "GOOD JOB!"
4990  RETURN

```



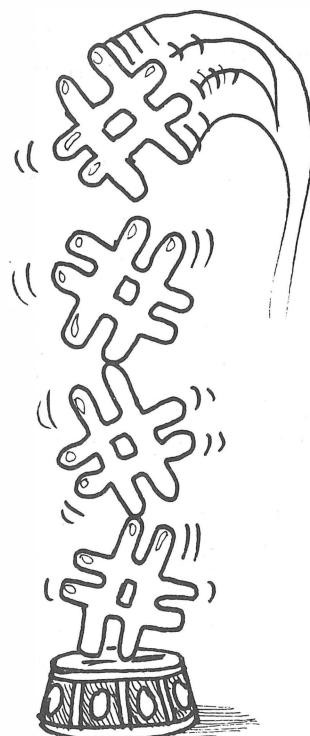


Impossible! It may seem impossible, but it's not. Deriving the key to this challenging game is very satisfying indeed! The object of the game is to transpose this list, # # # # . ? ? ? ? so that it looks like this ? ? ? ? . # # # #. The rules are few. The question marks (?) can only move to the left. Pound signs (#) can only move to the right. Either sign may be moved during a turn, with the following limitations. A sign may be moved into the place currently occupied by the period (this space is referred to as the blank space). A move is made by moving to an empty space or by jumping over one opposing piece. To understand the function of lines in the play section, separate compound lines and print out variables as you did for Reverser.

```

10 REM *****
11 REM *** ***
12 REM *** TRANSITION ***
13 REM *** ***
14 REM *****
15 REM
16 REM
20 GOSUB 1000: REM INSTS
30 GOSUB 2000: REM SETUP
40 GOSUB 3000: REM PLAY!
50 GOSUB 4000: REM !END!
60 END
1000 :
1001 REM *** INSTS
1002 :
1010 TEXT : NORMAL : HOME
1020 VTAB 3: HTAB 11: PRINT "***
      TRANSITION ***"
1030 VTAB 7: PRINT "THE GAME OF
      TRANSITION WILL PRESENT YOU
      WITH A LIST OF NINE DIGITS.
      THE LIST WILL LOOK LIKE T
      HIS : "
1039 PRINT
1040 PRINT "          1 2 3 4
      5 6 7 8 9
      # # # # , ? ? ? ?"
1050 PRINT : PRINT "THE OBJECT I
      S TO TRANSPOSE THE ORIGINAL
      CHARACTER POSITIONS. TRY TO
      REVERSE THE POUND SIGNS (#)
      AND THE QUESTION MARKS INTO
      ONE ANOTHER'S PREVIOUS POSI
      TIONS."
1060 VTAB 23: INPUT "HIT RETURN
      WHEN READY TO CONTINUE : ";A
      NS$
1070 HOME : VTAB 3: HTAB 11: PRINT
      "*** TRANSITION ***": VTAB 7
1080 PRINT "THE '#' CHARACTER CA
      N ONLY MOVE TO THE RIGHT, A
      ND THE '?' CHARACTER CAN ONL
      Y MOVE TO THE LEFT."

```



```

1090 PRINT : PRINT "A MOVE IS MA
DE BY MOVING TO AN EMPTY
SPACE, OR BY JUMPING OVER ON
E OPPOSING PIECE."
1100 PRINT : PRINT "TO MAKE A MO
VE, YOU ENTER THE POSITION
NUMBER OF THE MOVING PIECE.
TO QUIT, ENTER ZERO (0)."
```

1110 VTAB 23: INPUT "HIT RETURN  
WHEN READY TO CONTINUE : ";A  
NS\$

```

1990 RETURN
2000
2001 REM *** SETUP
2002
2010 DIM LI(9): FOR I = 1 TO 4:L
I(I) = 1:LI(10 - I) = 2: NEXT

2020 NM = 0
2990 RETURN
3000
3001 REM *** PLAY
3002
3010 HOME : VTAB 3: HTAB 11: PRINT
"*** TRANSITION ***": VTAB 6

3020 FOR I = 1 TO 9: IF (I < 5 AND
LI(I) = 2) OR (I = 5 AND LI(
I) = 0) OR (I > 5 AND LI(I) =
1) THEN NEXT :WL = 1: RETURN

3022 PRINT : PRINT "[";: FOR I =
1 TO 9: PRINT I;
3023 IF I < 9 THEN PRINT " ";:
3024 NEXT : PRINT "]" #----> <
---?"
3025 PRINT "[";: FOR I = 1 TO 9:
PRINT MID$ (",#?",LI(I) +
1,1);
3026 IF I < 9 THEN PRINT " ";:
3027 NEXT : PRINT " MOVE (0-
9) : "
```

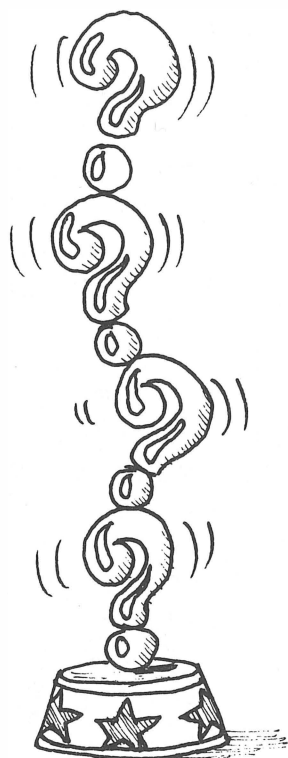
```

3030  GET ANS$: IF ANS$ < "0" OR
      ANS$ > "9" THEN 3030
3035  IF ANS$ = "0" THEN WL = 0: RETURN
3040  ANS = VAL (ANS$): PRINT ANS

3045  IF LI(ANS) = 0 THEN PRINT
      "THAT SPACE IS EMPTY ...": GOTO
      3020
3050  IF LI(ANS) = 1 THEN DI = 1
3051  IF LI(ANS) = 2 THEN DI = -
      1
3055  IF ANS + DI > 9 OR ANS + DI
      < 1 THEN PRINT "IT CANNOT
      MOVE FURTHER ...": GOTO 3020

3060  IF LI(ANS + DI) = 0 THEN LI
      (ANS + DI) = LI(ANS):LI(ANS)
      = 0:NM = NM + 1: GOTO 3020
3065  IF ANS + DI + DI > 9 OR ANS
      + DI + DI < 0 THEN PRINT "
      IT CANNOT MOVE FURTHER ...":
      GOTO 3020
3070  IF (LI(ANS + DI) < > LI(AN
      S)) AND (LI(ANS + DI + DI) =
      0) THEN LI(ANS + DI + DI) =
      LI(ANS):LI(ANS) = 0:NM = NM +
      1: GOTO 3020
3075  PRINT "IT CANNOT MOVE FURTH
      ER ...": GOTO 3020
3990  RETURN
4000  :
4001  REM *** END
4002  :
4010  PRINT : PRINT : PRINT "THE
      GAME IS OVER !!!": PRINT
4020  IF WL = 0 THEN PRINT "YOU
      GOT STUCK AFTER "NM" MOVES."
      : PRINT "BETTER LUCK NEXT TI
      ME !"
4025  IF WL = 1 THEN PRINT "YOU
      DID IT !!!": PRINT "AND IT O
      NLY TOOK "NM" MOVES."
4990  RETURN

```



For all of you word buffs, here is a game of anagrams geared towards any skill level. For those of you who are not familiar with anagrams, it is a word given in scrambled fashion. The following list should help.

SCRAMBLED    UNSCRAMBLED    RATING    = EASY 10 = VERY

xob	box	1 (elementary)
tahb	bath	2 (easy)
laott	total	3 (light)
betd	debt	4 (mild)
gindru	during	5 (moderate)
ptles	slept	6 (trying)
spumlie	impulse	7 (tough)
yrtaslc	crystal	8 (difficult)
meminscon	mnemonics	9 (hard)
preskulen	spelunker	10 (very hard)

In the actual game, the difficulty factor ranges from 1 to 5. The program does not utilize color, but there are some interesting points. You will note that there is a white border encircling each word. Line 3050 puts the computer into the INVERSE mode. Instead of plotting white onto black, the INVERSE is true, black will be plotted onto a white background. Lines 3050, 3060, and 3070 are responsible for drawing the white border. Copy these two short programs to see how the output is changed.

```
10 NORMAL 10 INVERSE
20 FOR A = 1 TO 10 20 FOR A = 1 TO 10
30 VTAB 2*I:HTAB I 30 VTAB 2*I:HTAB I
40 PRINT " " 40 PRINT " "
50 NEXT
```

Line 3080 is a loop which 'makes a pass' for each letter in the word (WL = Word Length). Although you see the entire word appear on the screen at one time, what is really happening is that one letter (WS\$) at a time is being printed. Line 3090 instructs the computer to PRINT CHR\$(95) once for every letter in the word. CHR(95) is a hyphen (-), so 3090 instructs the computer to print a hyphen for each letter in the word.

```

10 REM *****
11 REM *** ***
12 REM *** WORD SCRAMBLE ***
13 REM *** ***
14 REM *****
15 REM
16 REM
20 GOSUB 1000: REM INSTS
30 GOSUB 2000: REM SETUP
40 GOSUB 3000: REM PLAY!
50 GOSUB 4000: REM !END!
60 END
1000 :
1001 REM *** INSTS
1002 :
1010 TEXT : NORMAL : HOME
1020 VTAB 3: HTAB 9: PRINT "***
      WORD SCRAMBLE ***"
1030 VTAB 7: PRINT "IN THIS GAME
      , THE COMPUTER WILL CHOOSE
      A WORD AND SHOW YOU A SCRAMB
      LED VERSION."
1031 PRINT
1035 PRINT "YOUR PROBLEM IS TO U
      NSCRAMBLE THE WORD BEFORE T
      HE ALLOTTED TIME EXPIRES."
1040 VTAB 23: INPUT "HIT RETURN
      WHEN READY TO CONTINUE : ";A
      NS$
1990 RETURN
2000 :
2001 REM *** SETUP
2002 :
2010 HOME : VTAB 3: HTAB 9: PRINT
      "*** WORD SCRAMBLE ***"
2020 VTAB 7: PRINT "THE FOLLOWIN
      G ARE AVAILABLE : ": VTAB 10

2025 PRINT "      1) VERY EASY"
2026 PRINT "      2) EASY"
2027 PRINT "      3) INTERMEDIATE
      "
2028 PRINT "      4) HARD"
2029 PRINT "      5) VERY HARD"

```



```

2030  VTAB 18: INPUT "WHICH OPTIO
      N (1-5) : ";ANS
2035  IF ANS < 1 OR ANS > 5 OR AN
      S < > INT (ANS) THEN PRINT
      CHR$ (7): VTAB 18: CALL -
      958: GOTO 2030
2100  DIM WO$(100): FOR I = 1 TO
      100: READ WO$(I): NEXT
2105  DATA CAT,DOG,TREE,SIT,DOOR
      ,BOX,ARM,WALL,TEA,PEN,PAD,CU
      P,PIN,DIG,GOOD,TIE,SEA,ARE,H
      OW,LIP
2110  DATA WARM,WIND,LEAF,BLUE,W
      AIT,KITE,SLIP,DRIP,MAZE,PARK
      ,LIFE,GAME,HIGH,DISK,RUIN,CA
      RD,MOLE,ARCH,HARD,VERY,DATA
2115  DATA PENCIL,LOOSE,NORMAL,T
      IRED,BEFORE,AFTER,BLACK,TARG
      ET,KNOCK,BAGEL,INPUT,RETURN,
      START,ENTRY,GROUND,SHINE,HOR
      SE,PAPER,GREEN,PHONE
2120  DATA ORIGINAL,BEHIND,MAGAZ
      INE,STORAGE,SCRATCH,COMPUTER
      ,PERSONAL,SOFTWARE,PERFORM,S
      YSTEM,WINDOW,COMBINE,TANGENT
      ,SPECIFY,ANOTHER,EVALUATE,ME
      MORY,INSIDE,IGNORE,HOWEVER
2125  DATA SEQUOIA,MATRIX,COORDI
      NATE,SPACIAL,DIRECTION,SUBST
      ANTIAL,CONTINUE,SUBSCRIPT,EM
      ULATE,APPROPRIATE,CONICAL,DE
      VELOPMENT,ELEVATION,MECHANIC
      AL,MAGNETIC,TRAJECTORY,STIMU
      LUS,CIRCUMSTANCE,PROBABILITY
      ,PROJECTION
2200  DIM WR$(15),WS$(15),WC$(15)

2990  RETURN
3000  :
3001  REM *** PLAY
3002  :
3010  WO = INT ( RND (1) * 20) +
      (ANS - 1) * 20 + 1

```



```

3020 WL = LEN (WO$(WO)): FOR I =
      1 TO WL:WR$(I) = MID$ (WO$(
      WO),I,1):WS$(I) = WR$(I): NEXT

3025 FOR I = 1 TO WL:TI = INT (
      RND (1) * WL) + 1:WS$ = WS$
      (I):WS$(I) = WS$(TI):WS$(TI)
      = WS$: NEXT

3030 HOME : VTAB 3: HTAB 9: PRINT
      "*** WORD SCRAMBLE ***"

3040 VTAB 7: PRINT "HERE IS YOUR
      SCRAMBLED WORD : "

3050 VTAB 10: HTAB 10: INVERSE :
      FOR I = 1 TO WL + 4: PRINT
      " ";; NEXT

3060 VTAB 11: HTAB 10: PRINT " "
      ;; HTAB 10 + WL + 3: PRINT "
      ";; VTAB 12: HTAB 10: PRINT
      " ";; HTAB 10 + WL + 3: PRINT
      " ";; VTAB 13: HTAB 10: PRINT
      " ";; HTAB 10 + WL + 3: PRINT
      " ";

3070 VTAB 14: HTAB 10: FOR I = 1
      TO WL + 4: PRINT " ";; NEXT
      : NORMAL

3080 VTAB 12: HTAB 12: FOR I = 1
      TO WL: PRINT WS$(I);; NEXT

3090 VTAB 17: HTAB 12: FOR I = 1
      TO WL: PRINT CHR$ (95);; NEXT

3100 VTAB 20: HTAB 1: PRINT "(EN
      TER A-Z FOR THE LETTER,,, RE
      TURN WHENYOU ARE DONE.)"

3200 WP = 1

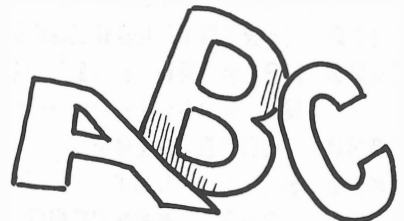
3205 FOR I = 250 * ANS TO 1 STEP
      - 1

3206 IF PEEK ( - 16384) > 127 THEN
      GOTO 3210

3207 IF INT (I / 50) * 50 = I THEN
      VTAB 12: HTAB 30: PRINT "TI
      ME: " INT (I / 50)" "

3208 NEXT : RETURN

```



```

3210  VTAB 17: HTAB 11 + WP: GET
      ANS$
3220  IF ASC (ANS$) = 8 THEN 330
      0
3230  IF ASC (ANS$) = 21 THEN 34
      00
3235  IF ASC (ANS$) = 13 THEN RETURN

3240  IF ANS$ < "A" OR ANS$ > "Z"
      THEN ANS$ = CHR$ (95)
3245  WC$(WP) = ANS$: PRINT ANS$;
3250  WP = WP + 1: IF WP > WL THEN
      WP = 1
3260  GOTO 3207
3300  IF WC$(WP) = "" THEN PRINT
      CHR$ (95);: GOTO 3320
3310  PRINT WC$(WP);
3320  WP = WP - 1: IF WP < 1 THEN
      WP = WL
3330  GOTO 3207
3340  IF WC$(WP) = "" THEN PRINT
      CHR$ (95);: GOTO 3420
3410  PRINT WC$(WP);
3420  WP = WP + 1: IF WP > WL THEN
      WP = 1
3430  GOTO 3207
4000  :
4001  REM *** END
4002  :
4010  FOR I = 1 TO WL: VTAB 12: HTAB
      11 + I: PRINT WR$(I);: IF WC
      $(I) = WR$(I) THEN INVERSE
      : VTAB 17: HTAB 11 + I: PRINT
      WC$(I);: NORMAL :WC = WC + 1

4020  NEXT : VTAB 20: HTAB 1: CALL
      - 958: IF WC = WL THEN PRINT
      "CONGRATULATIONS !": PRINT "
      YOU UNSCRAMBLD THE ENTIRE W
      ORD !": GOTO 4040
4030  PRINT "THE GAME IS OVER..."
      : PRINT "OUT OF "WL" LETTERS
      , YOU GOT "WC: PRINT "OF THE
      M CORRECT,"

```

```
4040 INPUT "DO YOU WISH TO PLAY  
AGAIN? ";ANS$: IF LEFT$ (AN  
S$,1) = "Y" THEN RUN  
4050 RETURN
```





This is an addicting game in which you try to consume all of the food squares before the Mubble Eaters consume you. A 9 by 9 arena houses the action. There are 64 food squares and three Mubble Eaters (ME's). If on the first run you do not complete the mission, you still have another Mubble to finish what the first Mubble started. It is up to you to safely guide the Mubble to gluttony. Go to it! This program will be explained line by line. It was chosen because the use of graphics is such that most of the other programs can be understood if this program is understood.

10 to 16 REM means remark. Anything may be construed as a remark. In this case, the REM statement is used to allow the program name to be written. Naturally, \*\*\*\*\* is not an actual REMark, but this statement (REM) allows the programmer to write the title in a computer-acceptable format.

15 and 16 The REM statements are followed by nothing. Their purpose is to maintain space between the title and line 20 (for aesthetic reasons).

20 GOSUB 1000 tells the computer to branch to line 1000 and to continue until the statement RETURN is encountered. When it is, the program will RETURN to line 20 and continue on to line 30. The colon which follows GOSUB 1000 is significant. .... A colon announces to the computer that a new instruction is forthcoming. An instruction which is preceded by a colon is exactly the same as an instruction preceded by a line number, with one important exception. If an instruction is a conditional (IF ..... THEN) the computer will perform the next sequential instruction if the condition is not met; but if the condition IS met, then every instruction on that line will be performed—even those instructions set off by a colon. What is present on lines 20-50 is a clumping of more than one instruction per line. You already know that a REM statement instructs the computer to ignore whatever follows it, so REM INSTS is written for your benefit, not the computer's. INSTS stands for INSTRUCTIONS.

30-50 are the same format as line 20. This is the key to the structuring of BASIC programs. All the instructions are in 1000-1999, setup at 2000-2999, etc.

60 When line 50 has been executed, the program is finished. END returns computer control to the user. You can have END anywhere in a program and even have it in several places as long as it does not get executed until the proper time!

1001 REM \*\*\* INSTS This line serves to inform the reader that the following lines contain the instructions.

1010 Line 1010 contains three instructions. TEXT is the instruction which changes the computer from the graphics (color) mode, back into the text (black and white) mode. NORMAL sets the the background color to black and the lettering color to white. HOME clears the screen (but only clears the screen of text, not color). It is standard practice to use this line in any program before a title screen is printed to 'clear the decks' of any text, graphics, or even garbage left over from a previous program.

1020 Line 1020 contains three instructions. VTAB 2 translates to Vertical TAB 2 lines. In other words, tab down two lines from the top of the screen. HTAB 10 translates to Horizontal TAB 10 spaces. This means, tab over ten spaces from the left-hand margin. PRINT says, output to the screen whatever is between the quotes. If you give the instruction: PRINT "X" then the output will be X. If you give the instruction: PRINT X then the numeric value contained in the variable 'X' will be printed. In the case of line 1020, the character string \*\*\* MUBBLE CHASE \*\*\* will be printed on the screen two lines down from the top and ten spaces in from the left margin.

1025 VTAB 5 This line instructs the computer to tab down five lines from the top of the page.

1030 This line PRINTs, verbatim, that which is between the quotes (five lines down from the top of the page).

1035 This line, in effect, PRINTs a blank line. The reason that this line is used is so that the text printed by line 1030 and the text to be printed by line 1040 will be separated by a blank line.

1050 This line contains two instructions. VTAB 23 tabs down 23 lines from the top of the page. The INPUT statement in this case serves only to hold up the program until you are ready to go on.

1060 This line contains five instructions. HOME clears the screen. VTAB 2 tabs down two lines from the top. HTAB 10 tabs over ten spaces from the left margin. PRINT outputs to the screen the material between the quotes, at a beginning position ten spaces from the left and two lines down from the top. After the PRINT statement is executed, the computer reads VTAB 5. This instruction says to skip down to the fifth line.

1070 This line PRINTs the material between the quotes on the fifth line from the top.

1090 (six instructions) HTAB 18 Horizontally TABs 18 spaces right. Next, the word 'UP' is printed. Then a blank line is printed (PRINT). Fourth, the computer is told to Horizontally TAB (to the right) 19 spaces. Then the letter 'T' is printed. Last, another blank line is printed below the 'T'.

1091 and 1092 finish the instructional chart which line 1090 began.

1100 This line is the same as line 1050.

1990 RETURN. The GOSUB 1000 in line 20 is completed by the required RETURN statement. Control is sent back to line 20, and then the program drops to line 30 which says to GOSUB 2000. This may seem a bit roundabout, but it is the basis of structuring the program into modular units.

2000 : This line and 2002 do nothing but aid in the readability of line 2001.

2001 REM \*\*\* SETUP This REMark tells you that the program setup is to follow.

2015 This instruction allows the variable ME to assume six different values (ME (1,1) ME (1,2) ME (2,1) ME (2,2) ME (3,1) ME (3,2), and the variable SP to assume three different values. The instruction DIM is short for DIMension. The computer is being told that ME is now a two-DIMensional array, and that SP will become a one-dimensional table. Instead of occupying a single location in memory, ME is now capable of occupying six, and SP can occupy three. A subscript (the numbers in the parentheses, ie. (1,1)) tells the computer where to find a certain value (in memory). Within a subscript there are rows and columns. The first value is always the length of each row (ME and SP both have rows which are three places long), and the second value (if present) is the length of each column. Another way to view subscripts is to envision the numbers as being: row,column. That is, ME(3,1) is located in the third row of the first column. Following will be an illustration of how the arrays are stored:

	<u>COLUMN 1</u>	<u>COLUMN 2</u>
ROW 1	ME (1,1)	ME (1,2)
ROW 2	ME (2,1)	ME (2,2)
ROW 3	ME (3,1)	ME (3,2)



This is a one dimensional array:

ROW 1	SP (1)	SP (2)	SP (3)
-------	--------	--------	--------

Actually, an array is assumed to have a storage location reserved for all zero subscripts, such as ME(0,0) ME(0,4) ME(3,0), but their use is normally omitted. Also, the computer does not actually store tables or arrays in two or three dimensions. The computer stores the data in one long string. The illustration is to help you visualize how to access various memory locations.

2030 PT is set to 0, and MU is set to 3 for reasons to be explained later.

2040 (two instructions) GR changes the mode from text (black and white) to GRaphics (color). HOME clears the bottom four lines which is the text area when in the GRaphics mode.

2041 This line specifies that all drawing is to be done in color 15 (white), until the color is changed.

2042 (two instructions) This line causes two Horizontal LINES to be printed (in white). Both lines travel from the left of the screen (0) to near the right edge (38, the lowest it could go is 39). The screen is broken down into 40 horizontal units (0-39) and 40 vertical units (0-39). When a horizontal line is to be printed, the computer needs to know which of the 40 rows to draw the line into. In the case of line 2042, the two lines are being drawn in row 0 and row 38.

2043 (two instructions) Line 2043 causes two Vertical LINES to be printed (still in white). This time the computer needs to know the column in which to draw the lines. The two lines are to be drawn in columns 0 and 38. What the result of lines 2040-2043 will be, is a white border traveling around the screen.

2050 This line changes the color of future drawings from 15 (white) to 1 (magenta).



2051 This line also bears close examination. The way to tell the computer to perform an action a certain number of times is by use of the FOR/NEXT loop. If you want a loop to be performed 6 times, the computer offers a number of ways to do this. The statement: FOR I = 1 TO 6 tells the computer to begin a loop with I equal to 1. Unless told otherwise, the value of I is incremented by one each time the loop is completed. After the sixth loop the value of I will change from 6 to 7. Since the FOR statement specifically said to perform the loop while I was equal to 1 through 6, the computer knows to stop looping when I = 7. Another way of telling the computer to loop six times would be: FOR X = 5 TO 10. By using the parameters 5 and 10, the loop will begin at X = 5 and continue until through X = 10. We could say FOR I = 1000 to 1005 and also achieve a loop to be performed six times. There are other ways to create a loop. If you write: FOR X = 3 TO 11, then the value of X is incremented by one each time the loop is performed. But if you write: FOR X = 3 TO 11 STEP 2, then X is incremented by TWO each time the loop is performed. For the first time through the loop X = 3, then two is added to X, so X = 5 for the second time through, then X = 7, X = 9, X = 11, and finally X = 13 (and the loop is done). This additional feature, STEP, merely allows the programmer to regulate the increment of the loop controller. Line 2051 is: FOR I = 2 TO 34 STEP 4. To interpret, this loop is performed when I = 2, 6, 10, 14, 18, 22, 26, 30, and 34. STEP 4 instructs the computer to increment X by 4, instead of by one, each time the loop is performed.

2052 What we have here is a loop within a loop, also referred to as a nested loop. This second loop (referred to as the J-loop) is performed each time the I-loop is performed.

2053 The I-loop gets further nested with the advent of the K-loop. Each time the I-loop is performed, the parameters of the K-loop will change.

2054 This line instructs the computer to draw a magenta line beginning at position J and drawing up to J + 2, and to draw the line at row K. Let's go back to line 2051 and see if we can follow this entire looping sequence. Line 2051 instructs the computer to perform the I-loop from 2 to 34 incrementing by 4. So I = 2. Line 2052 instructs the computer to perform the entire J-loop each time the I-loop is performed. J = 2 to 34 step 4, so to begin, J = 2. K = I to I + 2, so to begin (since I = 2), K = 2 to 4. Next, a Horizontal LINE is printed from J (2) to J+2 (4) at row K (2). Remember, the entire K-loop is performed with each pass of the J-loop. Continuing the K-loop, add one to K (K = 2 to 4, now K = 3). Draw a horizontal line from J (2) to J+2 (4) at row K (3). Completing the first of 81 K-loops, add one to K (K = 2 to 4, now K = 4). Draw horizontal line from 2 to 4 at 4. Now that the K-loop is done, the computer can continue the J-loop by incrementing J by 4. After doing this (line

2052) the computer is again instructed to perform the entire K-loop. The K-loop, remember, makes line 2054 get performed I to I + 2 times. Since I still equals 2, line 2053 can be rewritten as: FOR K = 2 to 4. Performing the K-loop for K = 2, 3, and 4, horizontal lines are drawn at 6 (J), 8 (J + 2) on row K. As you can see, the K-loop is entirely performed each time J is incremented. The J-loop is entirely performed each time the I-loop passes. To summarize, the I-loop is run just once, but has nine passes (a pass is an individual loop). The passes occur at 2, 6, 10, 14, 18, 22, 26, 30, and 34. With each pass of the I-loop the entire nine passes of the J-loop are performed. With each of the nine passes of the J-loop, all three passes of the K-loop are performed. A total of 243 passes of the K-loop are made. With each pass a horizontal line is drawn at J, J + 2 at row K. The 243 lines comprise the 81 magenta boxes you see on the screen.

2055 As mentioned before, a very easy and useful method of looping is by performing a FOR/NEXT loop. The FOR statement begins the loop (and each pass), conversely, the NEXT statement ends the loop (and each pass). When the NEXT statement is encountered, the computer will increment the variable as instructed. In the given example (line 2055), the variable K will be incremented each time a pass is made. The value of K starts at two and is incremented by one until it is equal to four. When the K-loop is started anew, K reverts to two. The variable J will be incremented each time that K = 4 (unless J = 34). The variable 'T' will be incremented only when K = 4 and J = 34. In other words, when an instruction line (such as line 2055) contains more than one NEXT variable, then the loop represented by the first one (K) is performed until completed. Then J is incremented, and again the K-loop is performed until completed. When J = 34 and the K-loop is done, only then is 'T' incremented. Each time that the 'T' variable is incremented, the J-loop starts with J = 2. The looping process continues until, on the two hundred forty-third pass, K = 4, J = 34, and I = 34.

2060 This line DIMensions the computer's memory to accept MU as a two-dimensional array. Six locations are reserved for MU values, not just one. Two rows with three columns each are set aside in the computer's memory.

2070 A FOR/NEXT loop is started here, with 'T' beginning at 5 and growing to 33 by increments of 4.

2071 Nested within the I-loop is a J-loop, also starting at J = 5 and continuing, by 4, until J = 33. The entire J-loop (eight passes) is performed each time the I-loop makes one of its eight passes.

2072 This line needs to be broken up into more digestible pieces. First, RND (1) will give a RaNDom number between 0 and 1. Actually, RND (7) also gives

you a random number between 0 and 1. Unless your computer has a special RND function, all random numbers are between zero and one. The random result is then multiplied by 4, giving a number between .00000004 and 3.99999996. Next, the number 6 is added to the total. At this point go back and look at the command INT. INT changes  $\text{RND}(1) * 4$  from a decimal into an INTeGer. This is done by truncating (chopping off) anything to the right of the decimal point. The number 3.996 becomes 3 (not 4). Because of this,  $\text{INT}(\text{RND}(1) * 4) + 6$  yields a random number between 6 and 9, not between 6 and 10. The result is that on each pass the color can be changed to 6, 7, 8, or 9.

2073 This line instructs the computer to plot a point, in the color given by the RaNDom INTeGer function, at X,Y coordinates I,J.

2074 As in line 2055, this line will increment J until  $J = 33$ , then 'I' will be incremented, J will revert to 5, and the looping procedure will continue until NEXT J,I is reached when both I and J are equal to 33. At that point, the computer will drop down to line 2990 (the next instruction).

2990 RETURN This line completes line 30, which instructs the computer to start a subroutine at line 2000, and to continue until the command RETURN is encountered.

3001 The REMark \*\*\* PLAY informs the reader that lines 3000-RETURN control the play.

3005 (nine instuctions) In this line, nine locations of ME are assigned values.

3006 (six instructions) First, the color is set to 2 (dark blue). Second, a loop is started. Third, the three Mubble Eaters are plotted onto the screen. Fourth, NEXT completes each pass of the loop. Because there is only one loop, the I-loop, the NEXT command needs no argument. Fifth, MD is set to 1. Sixth, the variable HI is set to 0.

3007 (twelve instructions) The first six instructions set values for the MUbble. Then values are given to the mubbles beginning X,Y coordinates, MX,MY. Remember, unlike a typical graph, the origin for the screen is in the UPPER left-hand corner, so the bottom left hand corner is at 0,39. The ninth instruction sets the color (of the MUbble) to 4. Next, a loop is started. The eleventh instruction plots the MUbble at the X,Y coordinates which are given. Finally, the NEXT command completes the I-loop.

3010 (two instructions) First, MU is decremented by one. Then a test is made ( $\text{MU} < 0$ ), and if the test (condition) proves to be true, the subroutine started by line 40 will be completed by RETURN.

3012 (six instructions) COLOR = 0 sets the color to 0 (black). A black, Horizontal LINE going all the way across the screen (from 0 to 39) to be drawn at row 39 is called for in the second instruction. Then the color is set to 15 (white). Next, a condition is made. If true, a loop is started, the point at (I \* 2), 39 is plotted each time a pass is made, and the NEXT statement marks the end of each pass.

3015 Remember, HOME only clears text, not color. In the GGraphics mode, only the bottom four lines are available for text. Next, a blank line is PRINTed; followed by the material between the quotes. The loop FOR I = 1 TO 2000 is merely a stalling tactic. The result of the loop is that the message <<< READY ..... >>> will stay on the screen while I is incremented by one, from 1 to 2000. This process takes from two to five seconds.

3020 Here we have a nested GOSUB. Line 40 initiated the subroutine beginning at line 3000. Now line 3020 instructs the computer to perform a subroutine within a subroutine. A REMark is made to explain the purpose of the subroutine at line 3300.

3025 This line is performed AFTER the subroutine beginning at line 3300. If you have eaten all 64 food points, then the subroutine (started by line 40) is completed.

3030 Here is another example of a nested GOSUB. The REMark tells us the purpose of the subroutine is to move the mubble eaters.

3040 'HI' is a special value. In most cases it is equal to "no" (which is "NOT HI"). If HI is equal to "yes", then, according to the program, the MUBble has been eaten by a Mubble Eater. If this is the case, then there is no need to go to 3020 (THEN 3020).

3041 To begin with, a loop is to be performed 60 times. In the loop, the variable XX is set to equal PEEK (-16336). This instruction (PEEK (-16336)) causes a clicking sound to be emitted from the speaker. This clicking sound is heard each time the Mubble eats one of the colored foodpoints. The NEXT statement concludes each pass.

3045 The FOR statement marks the start of a loop. The color is set to 0 (black), so when the Mubble Eaters move, after the Mubble moves, the positions on the maze where the Mubble Eaters were will not remain blue, but will be replaced with background colored points.

3050 This line draws the MUBble at its new position in the maze. This line is performed each time that the Mubble is moved.

3055 This line sends the program back to line 3005.

3300 Don't fret! Although this line appears to be a confusing conglomeration of variables, there is a definite purpose for this line. Before starting with an explanation, there are two important facts which you must know. One, MX and MY are the MUBbles X,Y coordinates. Two, the food points are located at specific intervals. Armed with this knowledge, you have a good chance of understanding what follows. The 64 food points are each located at an intersection. The X,Y coordinates at these 64 points are:

5,5	5,9	5,13	5,17	5,21	5,25	5,29	5,33
9,5	9,9	9,13	9,17	9,21	9,25	9,29	9,33
13,5	13,9	13,13	13,17	13,21	13,25	13,29	13,33
17,5	17,9	17,13	17,17	17,21	17,25	17,29	17,33
21,5	21,9	21,13	21,17	21,21	21,25	21,29	21,33
25,5	25,9	25,13	25,17	25,21	25,25	25,29	25,33
29,5	29,9	29,13	29,17	29,21	29,25	29,29	29,33
33,5	33,9	33,13	33,17	33,21	33,25	33,29	33,33

These points have one important thing in common. If you add three to any of the eight different X coordinates, the sum will be an exact multiple of four. Therefore,  $MX + 3$  divided by 4 will be an integer, and  $(MX + 3) / 4$  will be equal to INT (the integer value of)  $(MX + 3) / 4$ . Also, if you add three to any of the eight Y coordinates, then the sum will be evenly divisible by four; and  $(MY + 3) / 4$  will be equal to INT  $(MY + 3) / 4$ . The only times when both  $(MX + 3) / 4 = \text{INT}(MX + 3) / 4$  and  $(MY + 3) / 4 = \text{INT}(MY + 3) / 4$  is at one of the sixty-four intersections. If, indeed, the Mubble is at an intersection, then FL equals "yes". If the MUBble is not at one of the sixty-four intersections, then FL equals "no".

3301 This line is a conditional (a test). If FL is equal to "yes", then the computer will skip to line 3320.

3304 Each key on the keyboard has a corresponding numeric value referred to as an ASCII value. The keyboard ASCII values begin at 128 and proceed upward. PEEK (-16384) is an instruction which tells the computer to search the entire keyboard to see if any of the keys have been pressed. If they have, PEEK (-16384) will be equal to the ASCII value of whichever key was pressed. In other words, if any character on the keyboard was pressed, the value of PEEK (-16384) is going to be greater than 128.

3305 The variable KEY is set to the ASCII value of the key initially recognized by PEEK (-16384). As was mentioned before, the keyboard character ASCII

values begin at 128. For the sake of understanding, 128 is subtracted from PEEK (16384).

3310 This line checks to see if the KEY pressed was I. The ASCII value for 'I' is 73. If KEY is equal to 73 (I), then the variable MD (Mubble Direction) is set to 2. The computer is then told to branch to line 3319.

3311 This line checks to see if the KEY pressed was K. K is the game command which tells the Mubble to head west. If K was pressed, KEY is equal to 75, and the Mubble's Direction (MD) is set to 1.

3312 This line checks to see if the KEY pressed was M. The ASCII value for the letter M is 77. If, indeed, M was pressed, then Mubble Direction (MD) is set to 4 (undoubtedly this translates to MD = down).

3313 This line checks to see if the KEY pressed was J. If KEY equals 74, then the Mubble is going to head east. As in the previous three statements, the program will branch to line 3319 if the condition is met.

3315 This line will be performed only if KEY does not equal either 73 (I), 74 (J), 75 (K), or 77 (M). If none of the four conditions is true, then the program will bypass 3319 and go to 3320.

3319 This instruction, POKE 16368,0, rests the keyboard strobe, so that new information can be accepted from it. Simply, it clears the keyboard so that PEEK (-16384) can read new input, not continually reread the first key that was pressed.

3320 This line sets X2 equal to the Mubble's X coordinate, and Y2 equal to the Mubble's Y coordinate.

3321 If Mubble's Direction is equal to 1 (west), then add one to the X coordinate. If the mubble is headed west (right), then the value of the X coordinate increases with each move.

3322 If Mubble's Direction is equal to 2 (up), then subtract one from the Y coordinate. Because the origin is in the upper left-hand corner, the value of the Y coordinate increases as Y travels down the screen. To illustrate, as a point descends from the upper left-hand corner to the lower left-hand corner, the X,Y coordinates would look like this: 0,0 (at the origin), 0,1 0,2 0,3 0,4 0,5 0,6 and so on until, at the bottom of the screen we have 0,39.

3323 If Mubble's Direction is equal to 3 (east), then subtract one from the X

coordinate. As the mubble moves to the left (east), the value of the X coordinate is decremented.

3324 If Mubble's Direction is equal to 4 (south), then add one to the Y coordinate. As the mubble moves down (south), the value of Y is increased. This is due to the fact that the origin is in the upper left-hand corner.

3330 This line checks to see if either coordinate is out of the range of the maze. If it is, the subroutine is completed (RETURN), and the computer waits for you to input a viable keyboard character.

3337 It has already been determined that X2,Y2 are the X,Y coordinates of the mubble. SCRN returns the color value of the present (X,Y) cursor location. If the X,Y coordinates on the screen are equal to  $COLOR = 4$ . In other words, if, at coordinates X2,Y2, the SCReeN contains a mubble (the mubble is a greenish color ( $COLOR = 4$ )) then skip to line 3345.

3340 This line instructs the computer to check and make sure that the color at X2,Y2 is not black ( $SCRN\ X2,Y2 < > 0$  (black)), and if it is not black, the variable XX is set to be  $PEEK(-16336) + PEEK(-16336) - PEEK(-16336) + PEEK(-16336)$ . This equation may appear to be about as clear as a Chinese newspaper, but it is really not difficult to understand. Although XX appears to be set to equal a string of PEEK (-16336)'s, this is not the case. When the computer attempts to locate the value of PEEK (-16336), it is instructed to flick the toggle switch on the speaker. The result is that a short clicking sound is emitted. In the above equation, PEEK (-16336) appears four separate times, so the clicking sound is made four times. Because the clicks are emitted one right after the other, only someone with exceptional hearing can differentiate the one short click as being composed of four, shorter clicks. Remember, line 3300 ascertained that mubble was at a possible foodpoint location. Because the location was not black (line 3340) or mubble colored (line 3337), the location must have contained a foodpoint, and as a result, the player scores a Point  $PT = PT + 1$ ). Then the program checks to see if all 64 foodpoints have been scored. If they have, then the subroutine is completed by the RETURN command.

3345 This line is responsible for both moving the mubble ahead and replacing the mubble's last position with a black square. First, the color is set to 0 (black). Then, the hind third of the mubble's previous position, MU(1,1) and MU(1,2) is replaced by a black spot. But don't fret, the color is changed to the mubble's greenish color ( $COLOR = 4$ ), and the point where the mubble has moved to (the front one-third) is plotted. Last, the mubble's previous middle one-third becomes his rear one-third: MU(2,1 becomes MU(3,1), MU(2,1) becomes MU(2,2), MU(2,2) becomes MU(3,2).

3350 This line sets the new, front one-third coordinates into the memory locations MU(3,1) and MU(3,2). The new X coordinate for the mubble, X2, is now put into MX; and the new Y coordinate is moved into MY. is completes the task to be handled by this subroutine so it RETURNS.

3400 This line begins a loop which consists of three passes.

3402 Since there are three Mubble Eaters, the equation to check to see if they are at one of the 64 intersections has to be repeated three times. As with line 3300, if both conditions are met, then FL is, in effect, equal to 'yes'.

3405 With each pass of the loop, FL is roughly equivalent to 'yes' or to 'no'. If FL is 'no', then the program skips to line 3430.

3407 This line determines the Mubble Eaters' route of pursuit. The equation  $\text{INT}((\text{RND}(1) * 6) + 1)$  will yield an integer between 1 and 6. The function of the ON command is to send the program to the corresponding line number. What this means is, if the random number turns out to be one, then go to the FIRST line number listed (3410). If the random number is five, then the program will branch to the fifth line in the list of six (3430). There is an equal chance that the random number will be equal to 1,2,3,4,5, or 6. For each of the six possibilities, the program will branch to a certain line number. Following will be a list of the six random numbers, and the line number where the number will cause the program to branch.

1-3410, 2-3410, 3-3410, 4-3420, 5-3430, 6-3430

Each number has a one-in-six chance of being the random number, and because there are only three different line numbers in the list of six, there are different odds of branching to the three lines. If the random result is 1, 2, or 3, then the program will GO TO line 3410. There is a 50-50 chance (three in six) that this will happen. There is only a one-in-six chance that the the RaNDom number will be four, and the program will go to line 3420. The random results five and six both cause the program to GO TO line 3430. The odds of this are two-in-six.

3410 Another ON statement, which utilizes a random number function, is demonstrated in line 3410. The random result will be either one or two. If one, then the program will GO TO 3411; if the random result is two, then the program will branch to line 3413.

3411 Remember that the program is in a three-pass loop. This line compares the mubble's X coordinate to that of one of the three Mubble Eaters. If MX is



less than the Mubble Eater's X coordinate then the Mubble is to the left (east) of the Mubble Eater. Using the numbers 1 through 4 to indicate the four directions, 1 = west (right), 2 = north (up), 3 = east (left), and 4 = south (down), the value of ME(I,0) assumes the value of the direction (1-4) that the Mubble Eater should go.



3412 This line is the companion of line 3411. If the Mubble's X coordinate (MX) is greater than one of the three Mubble Eaters' X coordinate, then the mubble is west (left) of the ME's position, so ME(I,0) assumes the value which will later instruct the ME to move to the left. As in line 3411, if the condition is true ( $MX > ME(I,1)$ ) then the program branches to line 3430. Notice that if the Mubble Eater is on the same lateral plane (has the same X coordinate), that is, if the ME is on the same vertical line as the mubble, then the program falls through and starts trying to track the mubble by closing in on its Y coordinate (lines 3413 and 3414).

3413 Lines 3413 and 3414 have the same function as 3411 and 3412. The Y coordinate is the argument used for comparison. If the mubble's Y coordinate (MY) is smaller than the ME's Y coordinate, then the mubble is north (up), relative to the position of the ME.

3414 Likewise, if MY is greater than the Mubble Eater's Y coordinate, then the Mubble is south (down), relative to the position of the mubble eater.

3420 If the random number generated by line 3407 is four or if at 3413 MY is equal to ME(I,2), then line 3420 will be performed. This line gives the directional indicator (ME(I,0)) a random value of 1,2,3, or 4.

3430 Here the new coordinates for one of the ME's are set (using the information gained in lines 3411-3414).

3435 ME(I,0) is the directional indicator. To interpret, if ME(I,0) is equal to 1, then it is time to head east (right). This is accomplished by adding one to the ME's new X coordinate (X2).

3436 This time the line checks to see if the directional indicator says to "fly north." If so, this movement will be achieved by subtracting one from the ME's new Y coordinate.

3437 If ME(I,0) equals three, then the ME needs to travel west. This is done by summarily decrementing X2. If you find it difficult to visualize how this will result in the ME moving west, take out a piece of graph paper and experiment (keeping the origin in the upper left-hand corner.)

3438 Finally, if the mubble is south (down) in relation to the ME, then by increasing the value of the Mubble Eater's new Y coordinate, the ME will, indeed, move down.

3440 The function of this line, is to see if the ME's new X,Y coordinates (X2,Y2) are headed out of the maze boundaries. If they are, then the program branches to 3490 where corrective action will be taken.

3445 This instruction is very interesting. Its function is to remember what color the SPace the Mubble Eater is on was before it got there. The SCRN command reads the color off the screen and stores it in SP(I). Then, when the Mubble Eater has vacated its previous spot, the space's original color is restored. This way, the mubble eater does not leave a trail void of foodpoints, and it does not leave a dark blue trail either. To experiment, change line 3445 so that it reads: 3445 COLOR = 9, or 3445 COLOR = 2 or 3445 COLOR = 0. After making the change, run the program. Perhaps any misunderstandings will become clear.

3460 The foodpoints are plotted in the colors six through nine. This line checks to see if SP(I), which is equal to SCReen X2,Y2 (see 3447), is a foodpoint or not. If not, SP(I) reverts to 0.

3490 This line checks to see if a Mubble Eater has caught the MUBble. Because the MUBble occupies three spaces, the test has to verify the Mubble Eaters coordinates on the different points. If the coordinates of a Mubble Eater are the same as one of the three sets of MUBble coordinates, then the MUBble is done for. In this program, when the MUBble is eaten, recorded by setting HI (for HIIt) to one.

3495 The NEXT statement ends each pass of the loop begun on line 3400. When  $I = 3$ , this instruction ends the loop.

3990 RETURN completes the GOSUB. If this statement is omitted, the computer will stop the run to inform you of the error.

4000 This line merely serves to make 4001 more readable.

4001 This REMark tells us that the following lines comprise the \*\*\* END routine.

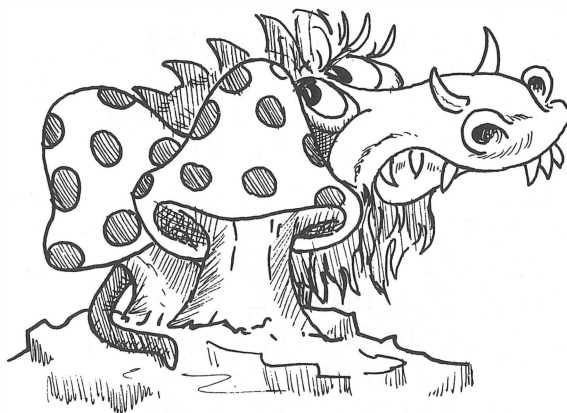
4002 A filler.

4010 (two instructions) HOME clears the four lines reserved for text. Next, the end-of-game message is printed.

4015 A test is run to see if you scored all 64 points. If you have, the message between the quotes is printed, and then the RETURN statement returns the program to line 50 and then line 60.

4020 The number of foodpoints you score is contained in location PT. Assuming 4015 was an invalid conditional, 4020 will print the exact message you have between the quotes, followed by the contents of the variable PT, and finally, the second half of the message which is between the quotes, will be printed precisely as it was written.

4990 If and when line 4020 is finished, this line RETURNs the program to line 50. Then the program continues on to line 60, where the flow is ended.



```

10  REM  *****
11  REM  ***                      ***
12  REM  *** MUBBLE CHASE ***
13  REM  ***                      ***
14  REM  *****
15  REM
16  REM
20  GOSUB 1000: REM  INSTS
30  GOSUB 2000: REM  SETUP
40  GOSUB 3000: REM  PLAY!
50  GOSUB 4000: REM  !END!
60  END
1000
1001  REM  *** INSTS
1002
1010  TEXT : NORMAL : HOME
1020  VTAB 2: HTAB 10: PRINT "***
      MUBBLE CHASE ***"
1025  VTAB 5
1030  PRINT "IN THIS EXCITING GAM
      E, YOU CONTROL THE  MOVEMENT
      OF THE HUNGRY LITTLE CREATU
      RE  WE CALL THE MUBBLE,  THE
      MUBBLE SCURRIESTHROUGH A MA
      ZE, TRYING TO EAT UP ALL OF
      THE FOOD POINTS."
1035  PRINT
1040  PRINT "UNFORTUNATELY, THERE
      ARE THREE MUBBLE-  EATERS I
      N THE SAME MAZE, WHO WANT TO
      CATCH AND EAT THE POOR M
      UBBLE."
1050  VTAB 23: INPUT "HIT RETURN
      WHEN READY TO CONTINUE : ";A
      NS$
1060  HOME : VTAB 2: HTAB 10: PRINT
      "*** MUBBLE CHASE ***": VTAB
      5
1070  PRINT "YOU MUST MANEUVER TH
      E MUBBLE TO THE FOODPOINTS A
      ND AWAY FROM THE MUBBLE EATE
      RS. YOU ARE ALLOWED TO LOSE
      TWO MUBBLES, BUTWHEN THE THI
      RD MUBBLE IS EATEN, THE GAME
      IS OVER."

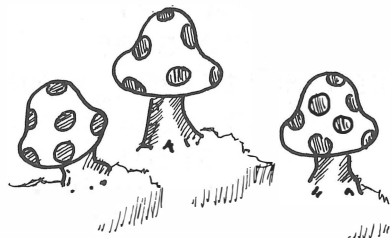
```

```

1075 PRINT
1080 PRINT "MOVEMENT OF THE MUBB
      LE IS CONTROLLED BY USING TH
      E LETTERS I, J, K, AND M,"
1090 HTAB 18: PRINT "UP": PRINT
      : HTAB 19: PRINT "I": PRINT

1091 PRINT "      LEFT  - J
      K - RIGHT": PRINT
1092 HTAB 19: PRINT "M": PRINT :
      HTAB 18: PRINT "DOWN"
1100 VTAB 23: INPUT "HIT RETURN
      WHEN READY TO CONTINUE : ";A
      NS$
1990 RETURN
2000 :
2001 REM *** SETUP
2002 :
2015 DIM ME(3,2),SP(3)
2030 PT = 0:MU = 3
2040 GR : HOME
2041 COLOR= 15
2042 HLIN 0,38 AT 0: HLIN 0,38 AT
      38
2043 VLIN 0,38 AT 0: VLIN 0,38 AT
      38
2050 COLOR= 1
2051 FOR I = 2 TO 34 STEP 4
2052 FOR J = 2 TO 34 STEP 4
2053 FOR K = 1 TO I + 2
2054 HLIN J,J + 2 AT K
2055 NEXT K,J,I
2060 DIM MU(3,2)
2070 FOR I = 5 TO 33 STEP 4
2071 FOR J = 5 TO 33 STEP 4
2072 COLOR= INT ( RND (1) * 4) +
      6
2073 PLOT I,J
2074 NEXT J,I
2990 RETURN
3000 :
3001 REM *** PLAY
3002 :

```



```

3005 ME(1,1) = 1:ME(1,2) = 1:ME(2
    ,1) = 37:ME(2,2) = 1:ME(3,1)
    = 37:ME(3,2) = 37:ME(1,0) =
    1:ME(2,0) = 4:ME(3,0) = 3
3006 COLOR= 2: FOR I = 1 TO 3: PLOT
    ME(I,1),ME(I,2): NEXT :MD =
    1:HI = 0
3007 MU(1,1) = 1:MU(1,2) = 37:MU(
    2,1) = 2:MU(2,2) = 37:MU(3,1
    ) = 3:MU(3,2) = 37:MX = 3:MY
    = 37: COLOR= 4: FOR I = 1 TO
    3: PLOT MU(I,1),MU(I,2): NEXT

3010 MU = MU - 1: IF MU < 0 THEN
    RETURN
3012 COLOR= 0: HLIN 0,39 AT 39:
    COLOR= 15: IF MU > 0 THEN FOR I=1
    TO
        MU: PLOT I * 2,39: NEXT I
3015 HOME : PRINT : PRINT "
        <<< READY ..... >>>": FOR
        I = 1 TO 2000: NEXT I: HOME

3020 GOSUB 3300: REM MOVE MUB
3025 IF PT = 64 THEN RETURN
3030 GOSUB 3400: REM MOVE MUB E
    ATERS
3040 IF NOT HI THEN 3020
3041 FOR I = 1 TO 60:XX = PEEK
    ( - 16336): NEXT
3045 FOR I = 1 TO 3: COLOR= SP(I
    ): PLOT ME(I,1),ME(I,2):SP(I
    ) = 0: NEXT
3050 PLOT MU(1,1),MU(1,2): PLOT
    MU(2,1),MU(2,2): PLOT MU(3,1
    ),MU(3,2)
3055 GOTO 3005
3065 MU(1,1) = 1:MU(1,2) = 37:MU(
    2,1) = 2:MU(2,2) = 37:MU(3,1
    ) = 3:MU(3,2) = 37:MX = 3:MY
    = 37: COLOR= 4: FOR I = 1 TO
    3: PLOT MU(I,1),MU(I,2): NEXT

```

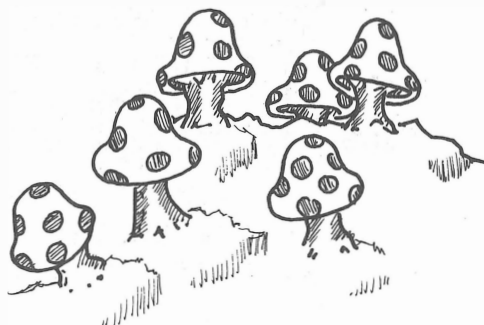
```

3300 FL = ((MX + 3) / 4) = INT (
      ((MX + 3) / 4)) AND ((MY + 3
      ) / 4) = INT ((MY + 3) / 4
      ))
3301 IF NOT FL THEN 3320
3304 IF PEEK ( - 16384) < 128 THEN
      3320
3305 KEY = PEEK ( - 16384) - 128

3310 IF KEY = 73 THEN MD = 2: GOTO
      3319
3311 IF KEY = 75 THEN MD = 1: GOTO
      3319
3312 IF KEY = 77 THEN MD = 4: GOTO
      3319
3313 IF KEY = 74 THEN MD = 3: GOTO
      3319
3315 GOTO 3320
3319 POKE - 16368,0
3320 X2 = MX:Y2 = MY
3321 IF MD = 1 THEN X2 = X2 + 1:
      GOTO 3330
3322 IF MD = 2 THEN Y2 = Y2 - 1:
      GOTO 3330
3323 IF MD = 3 THEN X2 = X2 - 1:
      GOTO 3330
3324 IF MD = 4 THEN Y2 = Y2 + 1:
      GOTO 3330
3330 IF X2 < 1 OR X2 > 37 OR Y2 <
      1 OR Y2 > 37 THEN RETURN
3337 IF SCRN( X2,Y2) = 4 THEN 3
      345
3340 IF SCRN( X2,Y2) < > 0 THEN
      XX = PEEK ( - 16336) + PEEK
      ( - 16336) - PEEK ( - 16336
      ) + PEEK ( - 16336):PT = PT
      + 1: IF PT = 64 THEN RETURN

3345 COLOR= 0: PLOT MU(1,1),MU(1
      ,2): COLOR= 4: PLOT X2,Y2:MU
      (1,1) = MU(2,1):MU(2,1) = MU
      (3,1):MU(1,2) = MU(2,2):MU(2
      ,2) = MU(3,2)

```



```

3350 MU(3,1) = X2:MU(3,2) = Y2:MX
    = X2:MY = Y2: RETURN
3400 FOR I = 1 TO 3
3402 FL = ((ME(I,1) + 3) / 4) = INT
    (((ME(I,1) + 3) / 4)) AND ((
    ME(I,2) + 3) / 4) = INT (((
    ME(I,2) + 3) / 4))
3405 IF NOT FL THEN 3430
3407 ON INT ( RND (1) * 6) + 1 GOTO
    3410,3410,3410,3420,3430,343
    0
3410 ON INT ( RND (1) * 2) + 1 GOTO
    3411,3413
3411 IF MX < ME(I,1) THEN ME(I,0
    ) = 3: GOTO 3430
3412 IF MX > ME(I,1) THEN ME(I,0
    ) = 1: GOTO 3430
3413 IF MY < ME(I,2) THEN ME(I,0
    ) = 2: GOTO 3430
3414 IF MY > ME(I,2) THEN ME(I,0
    ) = 4: GOTO 3430
3420 ME(I,0) = INT ( RND (1) * 4
    ) + 1: GOTO 3430
3430 X2 = ME(I,1):Y2 = ME(I,2)
3435 IF ME(I,0) = 1 THEN X2 = X2
    + 1: GOTO 3440
3436 IF ME(I,0) = 2 THEN Y2 = Y2
    - 1: GOTO 3440
3437 IF ME(I,0) = 3 THEN X2 = X2
    - 1: GOTO 3440
3438 IF ME(I,0) = 4 THEN Y2 = Y2
    + 1: GOTO 3440
3440 IF X2 < 1 OR X2 > 37 OR Y2 <
    1 OR Y2 > 37 THEN 3490
3445 COLOR= SP(I): PLOT ME(I,1),
    ME(I,2):SP(I) = SCRN( X2,Y2
    ): COLOR= 2: PLOT X2,Y2:ME(I
    ,1) = X2:ME(I,2) = Y2
3460 IF SP(I) < 6 THEN SP(I) = 0

```



```

3490 IF (ME(I,1) = MU(1,1) AND M
      E(I,2) = MU(1,2)) OR (ME(I,1
      ) = MU(2,1) AND ME(I,2) = MU
      (2,2)) OR (ME(I,1) = MU(3,1)
      AND ME(I,2) = MU(3,2)) THEN
      HI = 1
3495 NEXT : RETURN
3990 RETURN
4000 :
4001 REM *** END
4002 :
4010 HOME : PRINT "THE GAME IS O
      VER !!!"
4015 IF PT = 64 THEN PRINT "YOU
      GOT ALL THE POINTS (YOU WIN
      !!!)": RETURN
4020 PRINT "YOU SCORED "PT" POIN
      TS.....GOOD EFFORT"
4990 RETURN

```





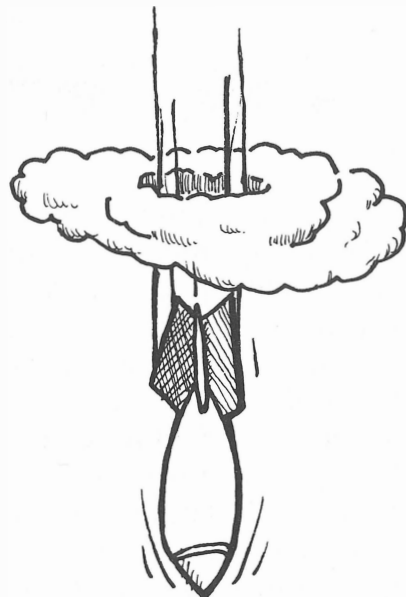
This game requires good timing. You are the pilot of a B19 Bomber trying to sink enemy ships. The graphics used in this game are simple but effective. To illustrate the point, run the program. The plane is green, the ship is purple, and the water is blue. In order to demonstrate how these three distinct forms are drawn, you will need to stop running the game (by typing Ctrl C). Type: LIST-3000. This command will list all of the program lines through (-) 3000, including 3000 (if it exists). Line 2060 sets the water to color = 2 (dark blue). Change this line to color = 6 or color = 12. After you are done experimenting, change 2060 back to its original configuration. Lines 2070 and 2080 instruct the computer where to draw the water, and how long to make it. To test this, change 2070 so it reads HLIN 0,20 AT 39, and then run the program. Likewise, experiment with line 2080. Next type: LIST-4000. To determine the function of lines 3100-3300, make changes in these statements and then run the program. The various subroutines are identified with remarks. If you would like to experiment with any of these routines, please do. Don't worry about making program changes. Your modified program will disappear when you turn off your computer or reload the program from the diskette. Unless you type: SAVE AIR ATTACK, none of the changes which you make will affect the program that is stored on the diskette. You should always save any modified versions you create under a new name such as AIR ATTACK 1, AIR ATTACK 2, or even 'JOE'S PROGRAM.'

```

10 REM *****
20 REM *** ***
30 REM *** AIR ATTACK ***
40 REM *** ***
50 REM *****
60 REM
70 REM
80 GOSUB 1000: REM INSTS
90 GOSUB 2000: REM SETUP
100 GOSUB 3000: REM PLAY!
110 GOSUB 4000: REM !END!
120 END
130 REM SL=SHOTS LEFT, YOU MAY
    MAKE A LONGER GAME BY GIVING
    MORE SHOTS IN LINE 2030

1000 :
1010 REM *** INSTS
1020 :
1030 TEXT : HOME : NORMAL
1040 VTAB 3: HTAB 11: PRINT "***
    AIR ATTACK ***"
1050 VTAB 7
1060 PRINT "IN THIS GAME YOU ARE
    A FIGHTER PILOT, YOU SCOR
    E BY HITTING ONE OF THE ENEM
    Y SHIPS WITH ONE OF YOUR B
    OMBS AND SINKING IT."
1070 PRINT
1080 PRINT "TO DROP A BOMB, SIMP
    LY PRESS ANY KEY ON THE KEYB
    OARD. YOUR SCORE FOR HITTIN
    G A SHIP WILL DEPEND ON WHIC
    H PART OF THE SHIP YOU HIT
    ,"
1090 PRINT
1100 PRINT "IF YOU HIT THE LOWER
    DECK, YOU SCORE 10 POINTS.
    IF YOU HIT THE UPPER DECK,
    YOU SCORE 20 POINTS. IF YOU
    HIT THE SMOKE- STACK YOU HA
    VE DONE VERY WELL, AND ARE
    REWARDED WITH 30 POINTS."
1110 VTAB 23

```



```

1120 INPUT "PRESS *RETURN* TO CO
      NTINUE : ";AN$
1130 HOME : VTAB 3: HTAB 11: PRINT
      "*** AIR ATTACK ***": VTAB 7

1140 PRINT "YOU HAVE AN ARSENAL
      OF 15 BOMBS. THE SPEED OF
      EACH SHIP WILL VARY, SO MAK
      E EVERY SHOT COUNT! "
1150 PRINT : PRINT "GOOD LUCK !!
      !"
1160 VTAB 23
1170 INPUT "PRESS *RETURN* TO CO
      NTINUE : ";AN$
1180 RETURN
2000
2010 REM *** SETUP
2020
2030 SL = 15
2040 REM LINE 2020 DRAWS THE W
      ATER
2050 GR
2060 COLOR= 2
2070 HLIN 0,39 AT 39
2080 HLIN 0,39 AT 38
2090 AX = 0: SX = 33: SS = 1
2100 RETURN
3000
3010 REM *** PLAY
3020
3030 HOME
3040 VTAB 22: CALL - 958: PRINT
      "SHOTS LEFT: "SL" SCORE:
      "TS
3050 GOSUB 3090: REM PLANE
3060 GOSUB 3130: IF SL = 0 THEN
      RETURN
3070 GOSUB 3270: REM SHIP
3080 GOTO 3050
3090 COLOR= 0: HLIN AX,AX + 6 AT
      2: HLIN AX + 1,AX + 6 AT 1: PLOT
      AX + 6,0

```

```

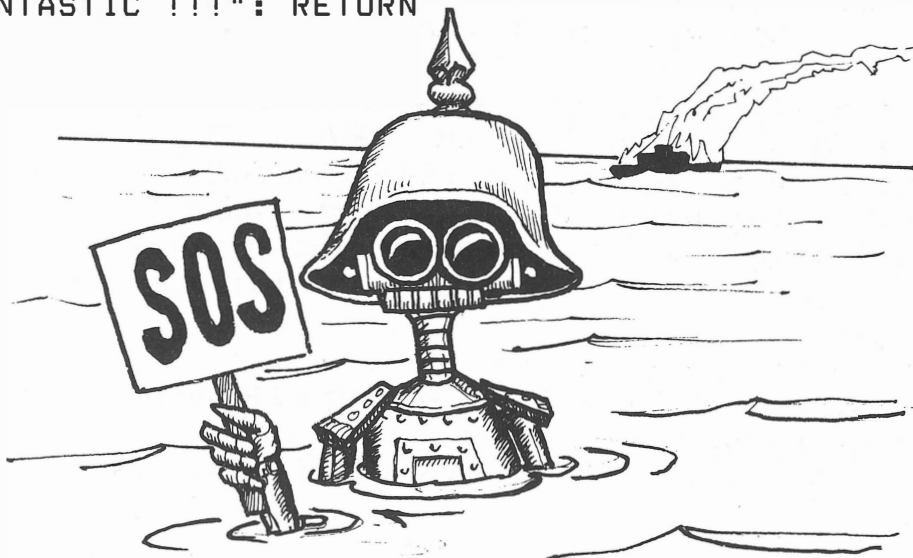
3100 REM LINE 3110 CHECKS TO S
    EEIF THE PLANE IS AT POSITIO
    N -1. IF SO, THE PLANE IS O
    FF THE SCREEN. AX REVERTS T
    O 33, THE RIGHT HAND SIDE O
    F THE PICTURE. FROM THERE T
    HE CYCLE BEGINS AGAIN.
3110 AX = AX - 1: IF AX =a 0 THEN
    AX = 33
3120 COLOR= 4: HLIN AX,AX + 6 AT
    2: HLIN AX + 1,AX + 6 AT 1: PLOT
    AX + 6,0: RETURN
3130 IF FF THEN 3160
3140 IF PEEK ( - 16384) =a 128 THEN
    RETURN
3150 FF = 1: POKE - 16368,0:FX =
    AX + 3:FY = 2
3160 COLOR= 0: PLOT FX,FY
3170 FY = FY + 1
3180 IF SCRN( FX,FY) = 0 THEN COLOR=
    13: PLOT FX,FY: RETURN
3190 IF SCRN( FX,FY) = 2 THEN 3
    230
3200 TS = TS + (38 - FY) * 10
3210 SC = 0: COLOR= 0: HLIN SX,SX
    + 6 AT 37: HLIN SX + 3,SX +
    5 AT 36: PLOT SX + 4,35
3220 SX = 33:SS = 1:SC = 0
3230 COLOR= 2: HLIN FX - 1,FX +
    1 AT 37: PLOT FX - 2,36: PLOT
    FX,36: PLOT FX + 2,36: PLOT
    FX - 3,35: PLOT FX,35: PLOT
    FX + 3,35
3240 COLOR= 0: HLIN FX - 1,FX +
    1 AT 37: PLOT FX - 2,36: PLOT
    FX,36: PLOT FX + 2,36: PLOT
    FX - 3,35: PLOT FX,35: PLOT
    FX + 3,35
3250 FF = 0:SL = SL - 1: POKE -
    16368,0
3260 VTAB 22: CALL - 958: PRINT
    "SHOTS LEFT: "SL" SCORE:
    "TS: RETURN

```

```

3270 SC = SC + 1: IF SC =a SS THEN
    RETURN
3280 SC = 0: COLOR= 0: HLIN SX,SX
    + 6 AT 37: HLIN SX + 3,SX +
    5 AT 36: PLOT SX + 4,35
3290 SX = SX + 1: IF SX =A 33 THEN
    SX = 0:SS = INT ( RND (1) *
    3) + 1:SC = SS
3300 COLOR= 1: HLIN SX,SX + 6 AT
    37: HLIN SX + 3,SX + 5 AT 36
    : PLOT SX + 4,35: RETURN
4000 :
4010 REM *** END
4020 :
4030 HOME : PRINT "THE GAME IS O
    VER ": PRINT "YOUR SCORE OF
    "TS" IS ";
4040 IF TS =a 25 THEN PRINT "ROT
    TEN !!!": RETURN
4050 IF TS =a 50 THEN PRINT "BAD
    !!!": RETURN
4060 IF TS =a 75 THEN PRINT "POO
    R !!!": RETURN
4070 IF TS =a 100 THEN PRINT "FA
    IR ...": RETURN
4080 IF TS =a 150 THEN PRINT "GO
    OD ...": RETURN
4090 IF TS =a 250 THEN PRINT "GR
    EAT !!!": RETURN
4100 IF TS =a 450 THEN PRINT "FA
    NTASTIC !!!": RETURN

```







To play Picasso, you must have a brush (paddle). It is always a good idea to play a game before you attempt the analysis. Load the program and type: LIST -1000. From this you will see where the subroutines begin. The first subroutine controls the messages and instructions you see before each run. To see how the program really works, list 3000-3040. To see how any instruction functions, type the line number with nothing after it. This deletes both the line and the line number. The lack of some attribute will reveal the line's purpose. Look at line 3003. This line instructs the computer to draw sixteen squares (points) at specific intervals. Line 3005 labels each color with a corresponding letter. Line 3020 merits an explanation. PEEK (-16384) instructs the computer to "survey" the keyboard. That is, the computer checks to see if any key has been pressed. Every keyboard character has a numeric equivalent. These numeric values are known as ASCII values and begin (for keyboard characters) at 128. Therefore, if PEEK (-16384) is less than 128, no key has been pressed. Line 3040 is a logical extension of 3020. The ASCII value for 'A' must be 193, and the ASCII value for 'P' must be 208. So, if the KEY that was pressed was A,B,C,D.....O,P, then the color is changed to the corresponding shade. If you type 3040 <RETURN>, the colors can no longer be chosen. Typing RUN ARTIST BOARD loads in a fresh copy of the program with everything working normally.

The noteworthy feature of the program loop (lines 3010 to 3090) is that it checks the game paddles and the keyboard at each pass. You can draw all day with the paddles but always be able to perform program functions by pressing

appropriate keys. You can use a loop like this in lots of games and other applications to allow for full use of the paddles and keyboard to control the program.

Lines 3110 and 3210 are interesting because they issue disk commands from a print statement by the use of CHR\$(4) which is CONTROL D. The BSAVE and BLOAD commands are followed by the location in memory which holds the contents of the low resolution graphics screen one.

Line 3010 is the heart of the program loop. The game paddles yield values from 0 to 255. This number has to be divided by a factor which will yield numbers from 0 to 39 since these are the limits of the low resolution screen. The INT function delivers whole numbers which are also required. After X and Y are read from the paddles a single small square is plotted in the color determined by line 1340. This color is then read off the screen by the SCRNB function and stored in 'C'. Next the color is changed to white (15) and the square replotted in white. Next comes a short delay loop to slow down the cycle and the blink rate. After this wait, the color is reassigned as 'C' and replotted. The effect is a square which marks the location of the paddles by blinking between white and the selected color. The selected color is generated before and after the white square so that you will always leave the right color behind even if you move the paddles fast. Try putting the parts of this line in a different order and see what a colorful hash results.

```
10 REM *****
11 REM ***
12 REM *** ARTIST BOARD ***
13 REM ***
14 REM *****
15 REM
16 REM
20 GOSUB 1000: REM INSTS
40 GOSUB 3000: REM PLAY!
50 GOSUB 4000: REM !END!
60 END
1000 :
1001 REM *** INSTS
1002 :
1010 TEXT : HOME : NORMAL
1020 VTAB 3: HTAB 10: PRINT "***
      ARTIST BOARD ***"
```

```

1030  VTAB 7: PRINT "BY USING ART
      IST BOARD AND YOUR CREATIVE
      TALENT, YOU CAN CREATE PICTU
      RES IN LOW- RESOLUTION COLOR
      .": PRINT
1040  PRINT : PRINT "THE GAME PAD
      DLES ARE USED TO MOVE THE
      COLORED CURSOR (YOUR PAINT B
      RUSH).  THE TOOLS ARE THERE
      FOR YOU TO DESIGN VERY ELAB
      ORATE PICTURES....GO TO IT !
      "
1042  PRINT : PRINT "YOU MAY SAV
      E A DRAWING ONTO DISK FOR
      RECALL AT A LATER TIME."
1050  VTAB 23: INPUT "HIT RETURN
      WHEN READY TO CONTINUE : ";A
      NS$
1990  RETURN
3000  :
3001  REM  *** PLAY!
3002  :
3003  GR : FOR I = 1 TO 16: COLOR=
      I: PLOT I * 2,39: NEXT
3005  HOME : PRINT "  A B C D E F
      G H I J K L M N O P": PRINT
      "TYPE IN CHOICE TO CHANGE CO
      LOR": PRINT "TYPE 'R' TO REC
      ALL, 'S' TO SAVE,": PRINT "'
      Q' TO QUIT, 'X' TO ERASE";
3010  X =  INT ( PDL (0) / 6.5):Y =
      INT ( PDL (1) / 6.7): PLOT
      X,Y:C =  SCRN( X,Y): COLOR=
      15: PLOT X,Y: FOR I = 1 TO 5
      0: NEXT I: COLOR= C: PLOT X,
      Y
3020  IF  PEEK ( - 16384) < 128 THEN
      3010
3030  KEY =  PEEK ( - 16384): POKE
      - 16368,0
3040  IF KEY >  = 193 AND KEY <  =
      208 THEN  COLOR= (KEY - 192)
      : GOTO 3010

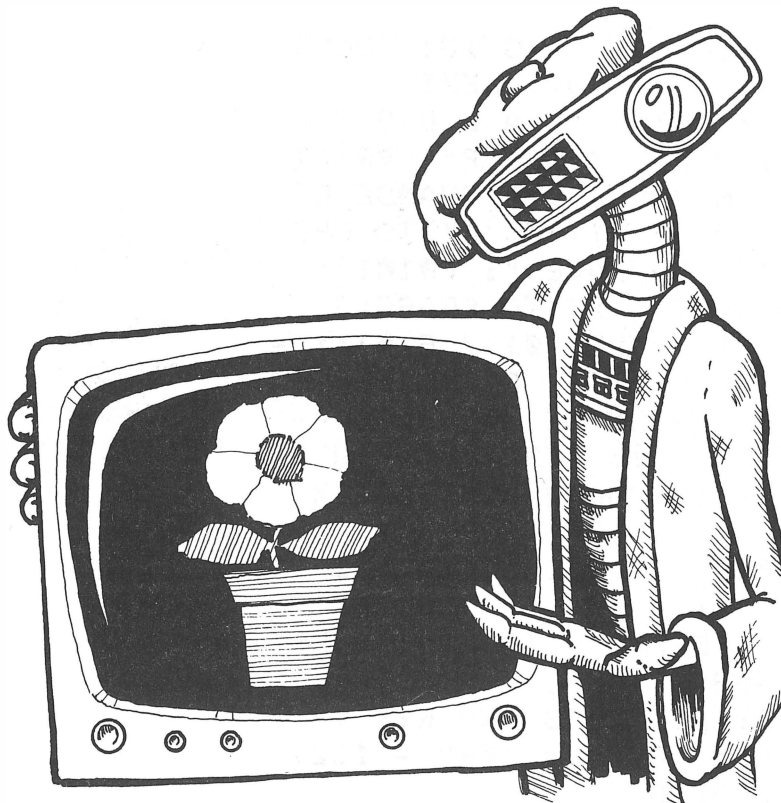
```

```

3050 IF KEY = 211 THEN 3100
3060 IF KEY = 210 THEN 3200
3070 IF KEY = 209 THEN RETURN
3080 IF KEY = 216 THEN 3003
3090 GOTO 3010
3100 HOME : INPUT "SAVE AS WHAT
FILE : ";FI$
3105 IF FI$ = "" THEN 3005
3110 PRINT CHR$ (4)"BSAVE "FI$
,A$400,L$400": GOTO 3005
3200 HOME : INPUT "RECALL WHAT F
ILE : ";FI$
3205 IF FI$ = "" THEN 3005
3210 PRINT CHR$ (4)"BLOAD "FI$
,A$400": COLOR= 0: GOTO 3005

3990 RETURN
4000 :
4001 REM *** !END!
4002 :
4010 TEXT : HOME : RETURN

```



This game is straightforward. Its use of colors and all graphics makes it a good exemplary program. Type: LOAD BARREL OF FUN, then list through line 2030. To see how each line works, change lines 2020-2029 as follows:

	2020 VTAB 19
omit	2021
do not change line	2022
	2023 COLOR = 1
	2024 FOR I = 4 TO 28 STEP 4
	2025 HLIN 8,30 AT I + 2
do not change	2026
	2027 FOR I = 4 TO 20 STEP 4
	2028 VLIN 6,28 AT I, do not change I.

Run the program after each change to see the new effect. These previously meaningless statements can be understood by using this technique. The following chart will clarify the function of each line:

LINE	COMMAND	FUNCTION
2020	VTAB 23	The computer tabs down 23 lines from the top of the screen
2021	CALL -958	Clears the text beginning at the cursor and continuing through to the bottom margin

2022	GR	Switches from the text mode (black and white) into the graphics mode (color)
2023	COLOR = 5	Until further notice, all printing will be done in gray (color=9)
2024	FOR I = 0 TO 18 STEP 6	Begins a loop which starts with I=0 with I being incremented by six each time a pass is completed
2025	HLIN 11,29 AT I	Draws a horizontal line from column 11 to column 29, on line I
2026	NEXT	Completes each pass of the loop begun at line 2024
2027	FOR I =0 to 18 STEP 6	(see line 2024)
2028	VLIN 0,36 AT I	Draws a vertical line beginning at row 0 and continuing to row 36. The line will be drawn at column I
2029	NEXT	Completes each pass of the loop begun at line 2027

```

10 REM *****
11 REM *** ***
12 REM *** BARREL OF FUN **
13 REM *** ***
14 REM *****
15 REM
16 REM
20 GOSUB 1000: REM INSTS
30 GOSUB 2000: REM SETUP
40 GOSUB 3000: REM PLAY!
60 END
1000 :
1001 REM *** INSTS
1002 :
1010 TEXT : NORMAL : HOME
1020 VTAB 3: HTAB 10: PRINT "***
    BARREL OF FUN ***"
1030 VTAB 7
1031 PRINT "IN BARREL OF FUN* YO
    U WILL SEE THREE COLUMNS
    OF SIX COLORFUL SQUARES. TH
    E OBJECT";
1032 PRINT " OF THE GAME IS TO A
    LIGN THE ROWS SUCH THAT EACH
    ROW IS ONE SOLID COLOR. TH
    ERE ARE ONLY TWO PURPLE SQUA
    RES. THE BLACK SQUARE MUST
    EVENTUALLY COMPLETE THE PU
    RPLE ROW."
1040 PRINT : PRINT "THE BOARD OF
    SQUARES WILL BE MIXED UP,
    AND YOUR TASK IS TO UNSCRAMB
    LE IT. THE FINAL PRODUCT SH
    OULD HAVE ALL OF THE SQUA
    RES OF ONE COLOR LINED UP IN
    A HORIZONTAL ROW.
1050 PRINT : PRINT "A SAMPLE OF
    HOW THE FINISHED PRODUCT
    SHOULD LOOK WILL BE SHOWN TO
    YOU. "
1060 PRINT : INPUT "HIT RETURN W
    HEN READY TO CONTINUE : ";AN
    S$

```



```

1070 HOME : VTAB 3: HTAB 10: PRINT
    "*** BARREL OF FUN ***": VTAB
    7
1080 PRINT "EACH COLUMN OF SQUAR
    ES (COLUMNS 1,2,3) CAN BE R
    OTATED VERTICALLY BY ENTERIN
    G THE NUMBER OF THE COLUMN
    YOU WISH TO ROTATE"
1090 PRINT : PRINT "YOU CAN MOVE
    COLORED SQUARES INTO THE
    EMPTY SQUARE BY USING THE AR
    ROWS ON THE KEYBOARD. IF TH
    E COLORED SQUARE IS TO BE M
    OVED TO THE EMPTY SQUARE ON
    ITS RIGHT, THEN PRESS TH
    E RIGHT ARROW. "
1100 PRINT : PRINT "IF THE COLOR
    ED SQUARE IS TO BE MOVED TO
    THE EMPTY SQUARE ON ITS LEFT
    , THEN PRESSTHE LEFT ARROW.
    "
1110 VTAB 23: INPUT "HIT RETURN
    WHEN READY TO CONTINUE : ";A
    NS$
1120 HOME : VTAB 3: HTAB 10: PRINT
    "*** BARREL OF FUN ***": VTAB
    7
1130 PRINT "NOTE THAT IN THE FIN
    AL SOLUTION ALL HORIZONTAL
    AL ROWS MUST BE THE SAME COL
    OR. "
1140 PRINT : PRINT "IT DOESN'T M
    ATTER WHERE THE ROW IS
    LOCATED, AS LONG AS ALL OF T
    HE COLORS WITHIN THE ROW A
    RE THE SAME. "
1150 VTAB 23: INPUT "HIT RETURN
    WHEN READY TO CONTINUE : ";A
    NS$
1990 RETURN
2000 :
2001 REM *** SETUP
2002 :

```

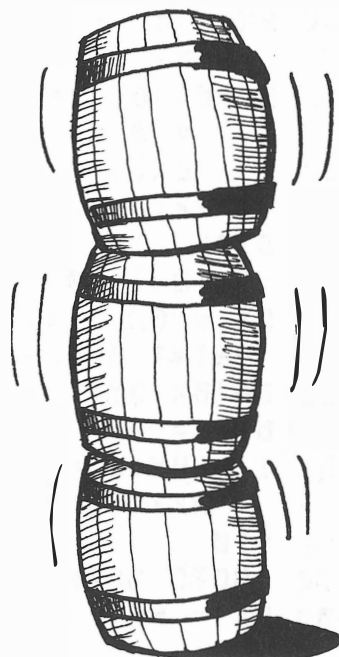


```

2010 DIM BA(3,6),CO(6): FOR I =
    1 TO 6: READ CO(I): NEXT : DATA
    1,8,11,12,15,3:BELL$ = CHR$
    (7)
2015 FOR I = 1 TO 6: FOR J = 1 TO
    3:BA(J,I) = CO(I): NEXT J,I:
    BA(3,6) = 0:BX = 3:BY = 6
2020 VTAB 23
2021 CALL - 958
2022 GR
2023 COLOR= 5
2024 FOR I = 0 TO 36 STEP 6
2025 HLIN 11,29 AT I
2026 NEXT
2027 FOR I = 0 TO 18 STEP 6
2028 VLIN 0,36 AT 1 + 11
2029 NEXT
2030 FOR I = 1 TO 3: GOSUB 2500:
    NEXT I
2100 HOME : HTAB 5: PRINT "<<< T
    HIS IS THE FINAL PATTERN >>>
    ": FOR I = 1 TO 2000: NEXT I

2110 HOME : HTAB 3: PRINT "<<< I
    'M NOW SCRAMBLING THE BOARD
    >>>"
2120 FOR N = 1 TO INT ( RND (1)
    * 10) + 20
2130 IF BX = 3 THEN RX = 2: GOTO
    2140
2131 IF BX = 1 THEN RX = 2: GOTO
    2140
2132 RX = INT ( RND (1) * 2) * 2
    + 1
2140 FOR L = 1 TO INT ( RND (1)
    * 5) + 1: FOR M = 1 TO 6:BA
    (RX,M - 1) = BA(RX,M): NEXT
    M:BA(RX,6) = BA(RX,0):I = RX
    : NEXT L: GOSUB 2500
2145 BA(BX,BY) = BA(RX,BY):BA(RX,
    BY) = 0
2150 I = BX:J = BY: GOSUB 2510
2155 BX = RX:I = BX:J = BY: GOSUB
    2510

```



```

2160 NEXT N: RETURN
2200 RETURN
2500 FOR J = 1 TO 6: GOSUB 2510:
    NEXT J: RETURN
2510 COLOR= BA(I,J): FOR K = 0 TO
    4: HLIN (I + 1) * 6,(I + 1) *
    6 + 4 AT (J * 6) - 5 + K: NEXT
    K: RETURN
2990 RETURN
3000 :
3001 REM *** PLAY
3002 :
3010 HOME : PRINT "(TYPE 1 TO RO
    TATE COLUMN 1, 2 FOR
    COLUMN 2, 3 FOR COLUMN 3, RI
    GHT ARROW )(TO SHIFT RIGHT,
    LEFT ARROW FOR LEFT )"
3020 IF PEEK ( - 16384) < 128 THEN
    3020
3025 KEY = PEEK ( - 16384): POKE
    - 16368,0
3030 IF KEY = 177 OR KEY = 178 OR
    KEY = 179 THEN 3100
3035 IF KEY = 136 THEN 3050
3040 IF KEY = 149 THEN 3075
3045 PRINT BELL$: GOTO 3010
3050 RX = BX + 1: IF RX = 4 THEN
    PRINT BELL$: GOTO 3010
3055 BA(BX,BY) = BA(RX,BY):BA(RX,
    BY) = 0
3060 I = BX:J = BY: GOSUB 2510:BX
    = RX:I = BX:J = BY: GOSUB 2
    510
3065 GOTO 3500
3075 RX = BX - 1: IF RX = 0 THEN
    PRINT BELL$: GOTO 3010
3080 BA(BX,BY) = BA(RX,BY):BA(RX,
    BY) = 0
3085 I = BX:J = BY: GOSUB 2510:BX
    = RX:I = BX:J = BY: GOSUB 2
    510
3090 GOTO 3500
3100 RX = KEY - 176

```

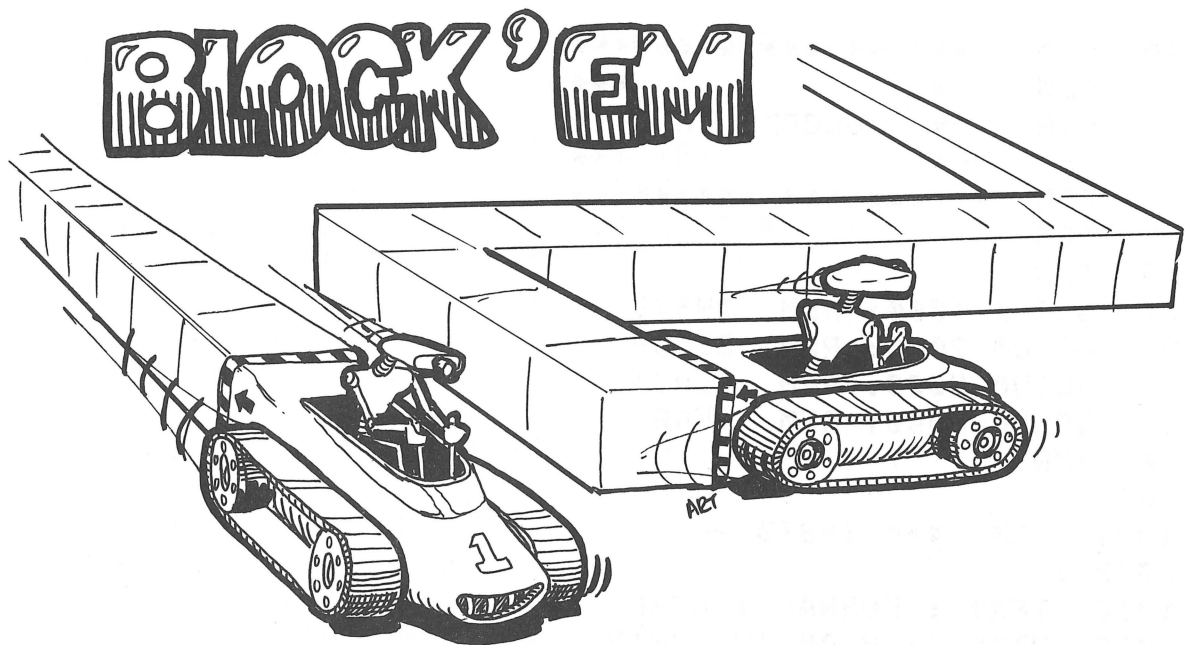
```

3110  FOR M = 1 TO 6:BA(RX,M - 1)
      = BA(RX,M): NEXT M:BA(RX,6)
      = BA(RX,0):I = RX: GOSUB 25
      00
3120  IF RX = BX THEN BY = BY - 1
      : IF BY = 0 THEN BY = 6
3125  GOTO 3500
3500  BA(BX,BY) = 3: FOR I = 1 TO
      6
3510  IF BA(1,I) = BA(2,I) AND BA
      (2,I) = BA(3,I) THEN NEXT I
      : RETURN
3520  BA(BX,BY) = 0:TRY = TRY + 1:
      GOTO 3010
3990  RETURN
4000  :
4001  REM *** END
4002  :
4010  HOME : PRINT BELL$BELL$"THE
      GAME IS OVER !!!": PRINT "Y
      OU DID IT IN "TRY" TRIES ...
      ": END
4990  RETURN

```







In this game you try to draw a longer line than your opponent. If your progress is impeded either by a border or by the opponent's line, you lose. This is a two man game. List through line 2125. The POKEs constitute an all-purpose sound routine which is explained in STARDODGER. Experiment with any of the lines on the screen. Line 2125 will be described in detail. HLIN stands for Horizontal LINE. A line has 40 characters (0-39), so the numbers 18,22 specify where the line will begin (at 18) and where the line will end (at 22). AT 18 specifies which line (on the Y, or vertical axis) receives the horizontal line. Next, another horizontal line is called for. It is to travel from 18 to 22 and is drawn on line 21. The colon ( : ) serves the same function as a new line and line number. In other words, a colon marks the end of one instruction and the beginning of another. To save space, many instructions may be clumped onto one line. Line 3010 draws the new position of both players. Change 3010 to read: 3010 COLOR = 1:PLOT X1,Y1:COLOR = 2:PLOT X2,Y2. Now when you run the program, the color of Player #1's line will be magenta (COLOR = 1) and the color of Player #2's line will be deep blue (COLOR = 2).

```

10 REM *****
11 REM *** ***
12 REM *** BLOCK 'EM ***
13 REM *** ***
14 REM *****
15 REM
16 REM
20 GOSUB 1000: REM INSTS
30 GOSUB 2000: REM SETUP
40 GOSUB 3000: REM PLAY!
50 GOSUB 4000: REM !END!
60 END
1000 :
1001 REM *** INSTS
1002 :
1010 TEXT : NORMAL : HOME
1020 VTAB 2: HTAB 12: PRINT "***
      BLOCK 'EM ***"
1030 VTAB 5
1031 PRINT "IN THIS GAME, TWO PL
      AYERS CONTROL THE CREATION
      OF A LINE."
1032 PRINT
1040 PRINT "THE FIRST PLAYER WHO
      SE LINE HITS A WALL,OR THE O
      THER PLAYER'S LINE,LOSES THE
      GAME. "
1050 PRINT : PRINT "PLAYER #1
      DIRECTION PLAYER #2": PRINT
1051 PRINT " W UP
      I": PRINT
1052 PRINT "A S LEFT
      RIGHT J K": PRINT
1053 PRINT " Z DOW
      N M"
1090 VTAB 23: INPUT "HIT RETURN
      WHEN READY TO CONTINUE : ";A
      NS$
1990 RETURN
2000 :
2001 REM *** SETUP
2002 :

```

```

2010 POKE 768,173: POKE 769,48: POKE
    770,192: POKE 771,136: POKE
    772,208: POKE 773,4: POKE 77
    4,198: POKE 775,1: POKE 776,
    240
2015 POKE 777,8: POKE 778,202: POKE
    779,208: POKE 780,246: POKE
    781,166: POKE 782,0: POKE 78
    3,76: POKE 784,0: POKE 785,3
    : POKE 786,96
2020 GR : HOME : COLOR= 15: HLIN
    0,39 AT 0: HLIN 0,39 AT 39: VLIN
    0,39 AT 0: VLIN 0,39 AT 39
2021 PRINT "PLAYER #1
        PLAYER #2"
2025 X1 = INT ( RND (1) * 5) + 1
    :Y1 = INT ( RND (1) * 10) +
    10:X2 = 38 - INT ( RND (1) *
    5):Y2 = INT ( RND (1) * 10)
    + 10:D1 = 1:D2 = 3
2030 FOR I = 1 TO 6: READ N,D: POKE
    0,N: POKE 1,D: CALL 768: NEXT
    I: DATA 110,75,70,75,55,75,4
    5,200,55,100,45,255
2100 FOR I = 5 TO 1 STEP - 1: COLOR=
    0: FOR J = 18 TO 22: VLIN 18
    ,24 AT J: NEXT : COLOR= 15: POKE
    0,200: POKE 1,5: CALL 768
2110 ON I GOTO 2121,2122,2123,21
    24,2125
2121 HLIN 18,22 AT 24: VLIN 18,2
    4 AT 20: HLIN 18,20 AT 18: GOTO
    2130
2122 HLIN 18,22 AT 18: HLIN 18,2
    2 AT 21: HLIN 18,22 AT 24: VLIN
    18,21 AT 22: VLIN 21,24 AT 1
    8: GOTO 2130
2123 HLIN 18,22 AT 18: HLIN 18,2
    2 AT 21: HLIN 18,22 AT 24: VLIN
    18,24 AT 22: GOTO 2130
2124 HLIN 18,22 AT 21: VLIN 18,2
    1 AT 18: VLIN 18,24 AT 22: GOTO
    2130

```

```

2125 HLIN 18,22 AT 18: HLIN 18,2
    2 AT 21: HLIN 18,22 AT 24: VLIN
    18,21 AT 18: VLIN 21,24 AT 2
    2: GOTO 2130
2130 FOR PA = 1 TO 300: NEXT PA,
    I
2135 COLOR= 0: FOR J = 18 TO 22:
    VLIN 18,24 AT J: NEXT : COLOR=
    15: POKE 0,200: POKE 1,5: CALL
    768
2990 RETURN
3000 :
3001 REM *** PLAY
3002 :
3010 COLOR= 7: PLOT X1,Y1: COLOR=
    9: PLOT X2,Y2
3015 NT = 5 + ((MO < 35) * (30 -
    INT (MO / 15) * 15))
3020 FOR I = 1 TO NT:KEY = PEEK
    ( - 16384): IF KEY < 128 THEN
    3050
3025 POKE - 16368,0
3030 IF KEY = 193 THEN D1 = 3
3031 IF KEY = 215 THEN D1 = 4
3032 IF KEY = 211 THEN D1 = 1
3033 IF KEY = 218 THEN D1 = 2
3040 IF KEY = 202 THEN D2 = 3
3041 IF KEY = 201 THEN D2 = 4
3042 IF KEY = 203 THEN D2 = 1
3043 IF KEY = 205 THEN D2 = 2
3050 NEXT I
3100 ON D1 GOTO 3110,3120,3130,3
    140
3110 X1 = X1 + 1: GOTO 3150
3120 Y1 = Y1 + 1: GOTO 3150
3130 X1 = X1 - 1: GOTO 3150
3140 Y1 = Y1 - 1: GOTO 3150
3150 IF X1 < 1 OR X1 > 38 OR Y1 <
    1 OR Y1 > 38 OR SCRN( X1,Y1
    ) < > 0 THEN WL = 2: RETURN

3200 ON D2 GOTO 3210,3220,3230,3
    240

```

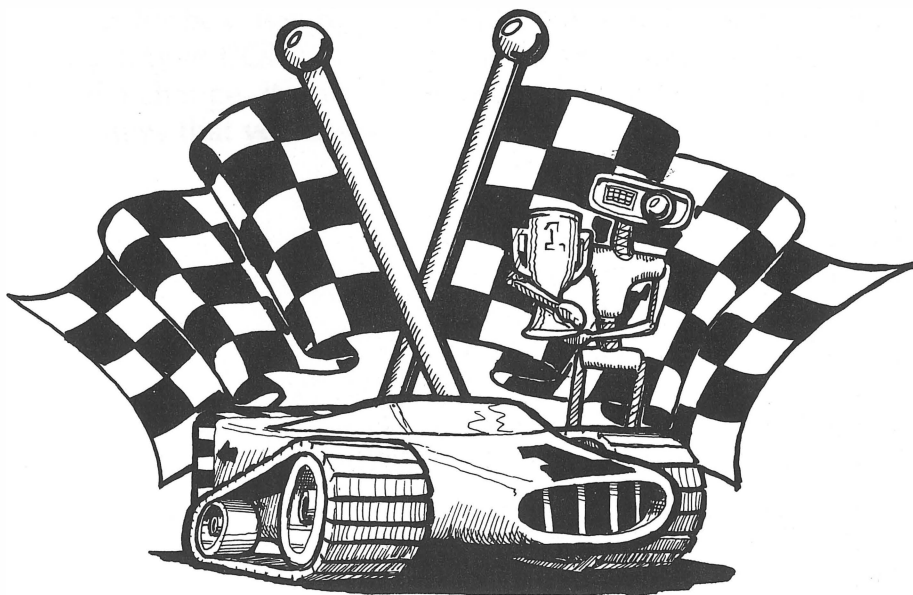


```

3210 X2 = X2 + 1: GOTO 3250
3220 Y2 = Y2 + 1: GOTO 3250
3230 X2 = X2 - 1: GOTO 3250
3240 Y2 = Y2 - 1: GOTO 3250
3250 IF X2 < 1 OR X2 > 38 OR Y2 <
    1 OR Y2 > 38 OR SCRN( X2,Y2
    ) < > 0 THEN WL = 1: RETURN

3300 POKE 0,90: POKE 1,30: CALL
    768:MO = MO + 1: GOTO 3010
4000 :
4001 REM *** END
4002 :
4010 HOME : PRINT "THE GAME IS O
    VER....": PRINT "PLAYER NUMB
    ER "WL" HAS WON THE GAME !!!
    "; CHR$ (7); CHR$ (7); CHR$
    (7)
4990 RETURN

```







This game is an intellectual challenge. Random selections will rarely net you a correct solution. It is a good idea to conceptualize how you intend to achieve your goal. Graphically, this program is straight-forward. Type: LIST-3030. Here are four lines (3010-3030) with which you may experiment. Line 3010 draws the original game board. Lines 3020-3030 have functions which are less apparent but just as important. If you look at lines 3505-3530, they help finish what line 3010 began. Verify the function of line 3525. It should draw the X in an occupied square. To check, type in a line 3522 and put STOP after the line number. Do likewise at line 3527. Now when the program is run, a break at 3522 will occur. When ready to continue, type: CONT. The new picture, which has an X in the center box, will be the result of line 3525. Again a break will occur (at line 3527); type CONT. To exit from the graphics mode, type: TEXT. Unless you SAVE a change, it won't be written to disk, so do not fret about undoing any changes that you make.

```

10 REM *****
11 REM *** ***
12 REM *** BRAIN TEASER ***
13 REM *** ***
14 REM *****
15 REM
16 REM
20 GOSUB 1000: REM INSTS
30 GOSUB 2000: REM SETUP
40 GOSUB 3000: REM PLAY!
50 GOSUB 4000: REM !END!
60 END
1000 :
1001 REM *** INSTS
1002 :
1010 TEXT : NORMAL : HOME
1020 VTAB 3: HTAB 10: PRINT "***
      BRAIN TEASER ***"
1030 VTAB 6
1031 PRINT "IN THIS GAME YOU ARE
      GIVEN A 3 BY 3 GAMEBOARD WI
      TH ONE OCCUPIED SPACE (THE
      CENTER), THE BOARD WILL
      RESEMBLE THIS : "
1040 PRINT : PRINT " - - -": PRINT
      " - X -": PRINT " - -
      -"
1045 PRINT
1050 PRINT "THE TRICK IS TO MOVE
      PIECES SO THAT THE GAME BOA
      RD WINDS UP LOOKING LIKE THI
      S:"
1055 PRINT
1060 PRINT " X X X
      AND - - - X -
      X NOT LIKE - - -
      X X X THIS:
      - - -"
1070 VTAB 23: INPUT "HIT RETURN
      WHEN READY TO CONTINUE : ";A
      NS$
1080 HOME : VTAB 3: HTAB 10: PRINT
      "*** BRAIN TEASER ***": VTAB
      7

```

```

1090 PRINT "YOU MAY ONLY MOVE TO
      AN OCCUPIED SPACE (A SPACE
      WITH AN X ON IT). WHEN YOU
      MOVE, CERTAIN SQUARES WILL
      REVERSE THEIR CONDITION (CH
      ANGE FROM AN X TO A BLANK,
      OR VICE-VERSA).
1095 PRINT
1100 PRINT "IF YOU MOVE TO A COR
      NER, ALL OF THE ADJACENT
      SQUARES REVERSE.
1102 PRINT "IF YOU MOVE TO THE M
      IDDL E OF A SIDE, ALL OF THE S
      QUARES ON THAT SIDE WILL FLI
      P, AND IF YOU CHOOSE THE CE
      NTER SQUARE, THAT BOX AND
      THE FOUR MIDDLE BOXES WILL
      ALL BE REVERSED"
1110 VTAB 23
1111 INPUT "HIT RETURN WHEN READ
      Y TO CONTINUE : ";ANS$
1120 HOME : VTAB 3: HTAB 10: PRINT
      "*** BRAIN TEASER ***": VTAB
      7
1130 PRINT "HERE IS A QUICK REVI
      EW OF THE VARIOUS MOVES, A
      ND THE RESULTING REVERSALS.,
      .,."
1135 PRINT
1140 PRINT "      M *          * M
      *      - * -"
1141 PRINT "      * * -      -
      * M *"
1142 PRINT "
      - * -"
1148 PRINT
1150 PRINT "THE 'M' DENOTES THE
      MOVE POSITION, AND THE '*'S
      DENOTE THE PIECES THAT WILL
      BE FLIPPED. THE PIECE AT '
      M' WILL ALSO BE FLIPPED. TH
      E BOARD IS DESIGNATED LIKE
      THIS : "

```

```

1160 PRINT "          1 2 3": PRINT
    "          4 5 6": PRINT "
          7 8 9"
1170 VTAB 23
1171 INPUT "PRESS RETURN WHEN RE
ADY TO CONTINUE : ";ANS$
1990 RETURN
2000 :
2001 REM *** SETUP
2002 :
2010 DIM BO(3,3): FOR I = 1 TO 3
    : FOR J = 1 TO 3:BO(I,J) = -
    1: NEXT J,I:BO(2,2) = 1
2990 RETURN
3000 :
3001 REM *** PLAY
3002 :
3010 GR : HOME : FOR I = 1 TO 3:
    FOR J = 1 TO 3: GOSUB 3505:
    NEXT J,I
3020 FOR I = 1 TO 3: FOR J = 1 TO
3: IF BO(I,J) = 1 OR (I = 2 AND
J = 2) THEN NEXT J,I:WL = 1
: RETURN
3025 FOR I = 1 TO 3: FOR J = 1 TO
3: IF BO(I,J) = - 1 THEN NEXT
J,I:WL = 0: RETURN
3030 HOME : PRINT "INPUT A POSIT
ION (1-9) OR": PRINT "ENTER
RETURN TO QUIT ==> ";; INVERSE
: PRINT " ";; NORMAL
3040 IF PEEK ( - 16384) < 128 THEN
3040
3050 KEY = PEEK ( - 16384): POKE
- 16368,0
3060 IF KEY = 141 THEN RETURN
3070 IF KEY < 177 OR KEY > 185 THEN
3030

```

```

3075 KEY = KEY - 176:I = INT ((K
    EY - 1) / 3) + 1:J = KEY - (
    I - 1) * 3: IF B0(I,J) =
    1 THEN HOME : PRINT "MOVE O
    NLY TO AN OCCUPIED SQUARE": FOR
    I = 1 TO 1500: NEXT : GOTO 3
030
3080 ON KEY GOTO 3110,3120,3130,
    3140,3150,3160,3170,3180,319
0
3110 I = 1:J = 1: GOSUB 3500
3111 I = 1:J = 2: GOSUB 3500
3112 I = 2:J = 1: GOSUB 3500
3113 I = 2:J = 2: GOSUB 3500: GOTO
    3020
3120 I = 1:J = 1: GOSUB 3500
3121 I = 1:J = 2: GOSUB 3500
3122 I = 1:J = 3: GOSUB 3500: GOTO
    3020
3130 I = 1:J = 2: GOSUB 3500
3131 I = 1:J = 3: GOSUB 3500
3132 I = 2:J = 2: GOSUB 3500
3133 I = 2:J = 3: GOSUB 3500: GOTO
    3020
3140 I = 1:J = 1: GOSUB 3500
3141 I = 2:J = 1: GOSUB 3500
3142 I = 3:J = 1: GOSUB 3500: GOTO
    3020
3150 I = 1:J = 2: GOSUB 3500
3151 I = 2:J = 1: GOSUB 3500
3152 I = 2:J = 2: GOSUB 3500
3153 I = 2:J = 3: GOSUB 3500
3154 I = 3:J = 2: GOSUB 3500: GOTO
    3020
3160 I = 1:J = 3: GOSUB 3500
3161 I = 2:J = 3: GOSUB 3500
3162 I = 3:J = 3: GOSUB 3500: GOTO
    3020
3170 I = 2:J = 1: GOSUB 3500
3171 I = 2:J = 2: GOSUB 3500
3172 I = 3:J = 1: GOSUB 3500
3173 I = 3:J = 2: GOSUB 3500: GOTO
    3020

```

```

3180 I = 3:J = 1: GOSUB 3500
3181 I = 3:J = 2: GOSUB 3500
3182 I = 3:J = 3: GOSUB 3500: GOTO
    3020
3190 I = 2:J = 2: GOSUB 3500
3191 I = 2:J = 3: GOSUB 3500
3192 I = 3:J = 2: GOSUB 3500
3193 I = 3:J = 3: GOSUB 3500: GOTO
    3020
3500 BO(I,J) = - BO(I,J)
3505 COLOR= BO(I,J) + 5
3510 FOR I2 = I * 8 TO I * 8 + 7
    : HLIN J * 8,J * 8 + 7 AT I2
    : NEXT I2
3520 IF BO(I,J) = - 1 THEN COLOR=
    15: FOR I2 = I * 8 + 3 TO I *
    8 + 4: HLIN J * 8 + 3,J * 8 +
    4 AT I2: NEXT : RETURN
3525 COLOR= 15: FOR I2 = I * 8 +
    1 TO I * 8 + 6: PLOT ((J * 8
    ) + (I2 - (I * 8))),I2: PLOT
    ((J * 8) + 7 + ((I * 8) - I2
    )),I2: NEXT
3540 RETURN
4000 :
4001 REM *** END
4002 :
4010 HOME : PRINT CHR$ (7) CHR$
    (7) CHR$ (7)"THE GAME IS OVE
    R !!!"
4020 IF WL = 0 THEN PRINT "AND
    YOU'VE LOST...SORRY "
4030 IF WL = 1 THEN PRINT "AND
    YOU'VE WON....GREAT!!!"
4990 RETURN

```

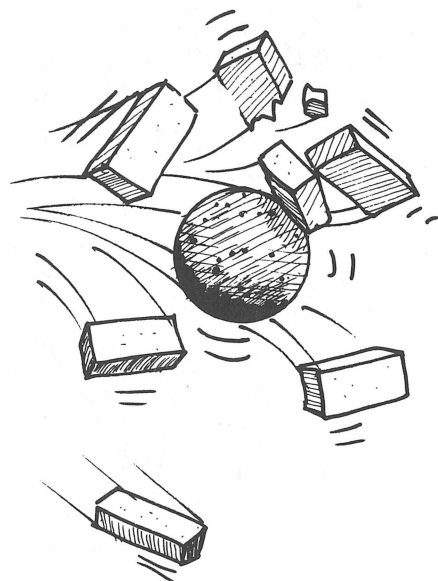


What a challenge! This game tests your quickness and dexterity. The game is similar to some of the arcade games you see. With a little practice YOU could write this program. Let's get to the heart of the program. First, you must LOAD BRICK WALL. You should RUN the program to get the feel of it. Now type: LIST -1960. Look at line 2010. To get into the GGraph mode (for color) there must be the command GR (for graphics). To dramatize the function of a particular line, type in the line number and return; this deletes the line. Then run. What goes wrong? Does the program crash or just lose some functions? To see the actual workings of the program, type LIST. What you see is a roadmap telling you where to find various routines. Again, if you would like to know the function of a line, type in the line number, return, and run. Some of the lines that you should experiment with are: 2110, 2310, 2331 and 3020.

```

10 REM *****
11 REM *** ***
12 REM *** BRICK WALL ***
13 REM *** ***
14 REM *****
15 REM
16 REM
20 GOSUB 1000: REM INSTS
30 GOSUB 2000: REM SETUP
40 GOSUB 3000: REM PLAY!
50 GOSUB 4000: REM !END!
60 END
1000 :
1001 REM *** INSTS
1002 :
1010 TEXT : NORMAL : HOME
1020 VTAB 3: HTAB 12: PRINT "***
    BRICK WALL ***"
1030 VTAB 7: PRINT "IN THIS GAME
    , YOU WILL BE PRESENTED
    WITH A WALL OF BRICKS AT THE
    TOP OF THE SCREEN, AND A PA
    DDLE AT THE BOTTOM. THE
    GAME PADDLE (PADDLE 0) IS US
    ED TO HIT A ROCK INTO THE
    WALL OF BRICKS.": PRINT
1031 PRINT "WHEN THE ROCK HITS A
    BRICK, IT WILL DESTROY
    IT AND POINTS WILL BE ADDED
    TO YOUR SCORE."
1040 PRINT : PRINT "YOUR MISSION
    , SHOULD YOU ACCEPT, IS TO
    DESTROY AS MUCH OF THE BRICK
    WALL AS POSSIBLE.": PRINT
    : PRINT "YOU ARE ALLOWED ONL
    Y 5 MISSES."
1060 VTAB 23: INPUT "HIT RETURN
    WHEN READY TO CONTINUE : ";A
    NS$
1990 RETURN
2000 :
2001 REM *** SETUP
2002 :
2010 GR

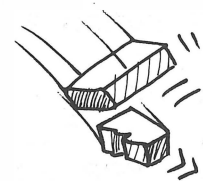
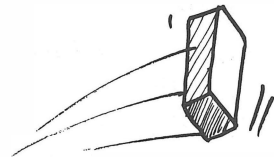
```



```

2020 GOSUB 2100
2025 GOSUB 2400
2030 DIM DIR(6): FOR I = 1 TO 6:
    READ DIR(I): NEXT I: DATA -
    1.5,-1,-.5,.5,1,1.5
2090 RETURN
2100 REM *** DO BACKGROUND
2110 FOR I = 5 TO 19 STEP 2: K =
    (I - 1) / 2: K = K - (INT (K
    / 2) * 2)
2115 COLOR= K * 4 + 9: FOR J = 0
    TO 36 STEP 4: HLINE J,J + 1 AT
    I: NEXT J
2120 COLOR= (1 - K) * 4 + 9: FOR
    J = 2 TO 38 STEP 4: HLINE J,J
    + 1 AT I: NEXT J
2125 NEXT I
2190 RETURN
2200 REM *** DRAW PADDLE
2201 P = ( PDL (0) - 20) / 6
2202 IF P < 0 THEN P = 0
2203 IF P > 34 THEN P = 34
2210 IF P = PP THEN RETURN
2215 COLOR= 0: HLINE PP,PP + 5 AT
    39: COLOR= 6: HLINE P,P + 5 AT
    39: PP = P: RETURN
2300 REM *** MOVE BALL
2305 X2 = BX + DI(BA): Y2 = BY + B
    D
2310 IF X2 < 0 OR X2 > 39 THEN X
    2 = BX - DI(BA): BA = 7 - BA:
    POKE 0,220: POKE 1,10: CALL
    768
2315 IF Y2 < 0 THEN Y2 = BY - BD
    :BD = - BD:BF = 1: POKE 0,2
    00: POKE 1,10: CALL 768
2320 IF Y2 > 39 THEN COLOR= 0: PLOT
    BX,BY: POP : GOTO 3060
2325 IF SCRN( X2,Y2) < > 6 THEN
    2330
2326 POKE 0,240: POKE 1,10: CALL
    768:BD = - BD:BF = 0:BA = INT
    (X2) - INT (PP) + 1:PB = PB
    + 1: IF PB = 7 THEN BD = -
    2

```



```

2327  COLOR= 0: PLOT BX,BY: COLOR=
      15: PLOT X2,Y2: COLOR= 6: PLOT
      X2,Y2:BX = X2:BY = Y2: RETURN

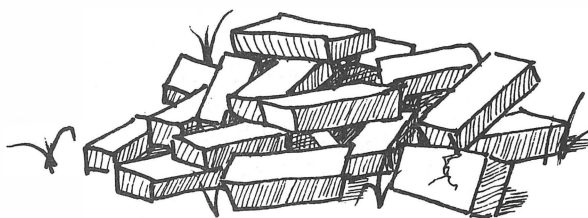
2330  IF  SCRN( X2,Y2) = 0 THEN 2
      335
2331  COLOR= 0:X3 = ( INT (X2 / 2
      ) * 2): HLIN X3,X3 + 1 AT Y2
      :SC = SC + (10 - (Y2 - 1) /
      2): POKE 0,10: POKE 1,10: CALL
      768
2332  IF SC = INT (SC / 720) * 7
      20 THEN POP : RETURN
2333  IF BD < 0 OR BF = 1 THEN BD
      = - BD
2334  VTAB 22: HTAB 9: PRINT SC
2335  IF  SCRN( BX,BY) <  > 15 THEN
      2345
2340  COLOR= 0: PLOT BX,BY
2345  COLOR= 15: PLOT X2,Y2:BX =
      X2:BY = Y2: RETURN
2400  REM  *** MUSIC TONES
2405  POKE 768,173: POKE 769,48: POKE
      770,192: POKE 771,136: POKE
      772,208: POKE 773,4: POKE 77
      4,198: POKE 775,1: POKE 776,
      240
2410  POKE 777,8: POKE 778,202: POKE
      779,208: POKE 780,246: POKE
      781,166: POKE 782,0: POKE 78
      3,76: POKE 784,0: POKE 785,3
      : POKE 786,96: RETURN
3000  :
3001  REM  *** PLAY
3002  :
3005  HOME : VTAB 22: PRINT "SCOR
      E : "
3006  VTAB 21: PRINT "BALLS LEFT
      : "
3010  FOR I = 1 TO 5: VTAB 21: HTAB
      14: PRINT 5 - I
3015  FOR J = 1 TO 100: GOSUB 220
      0: NEXT J

```

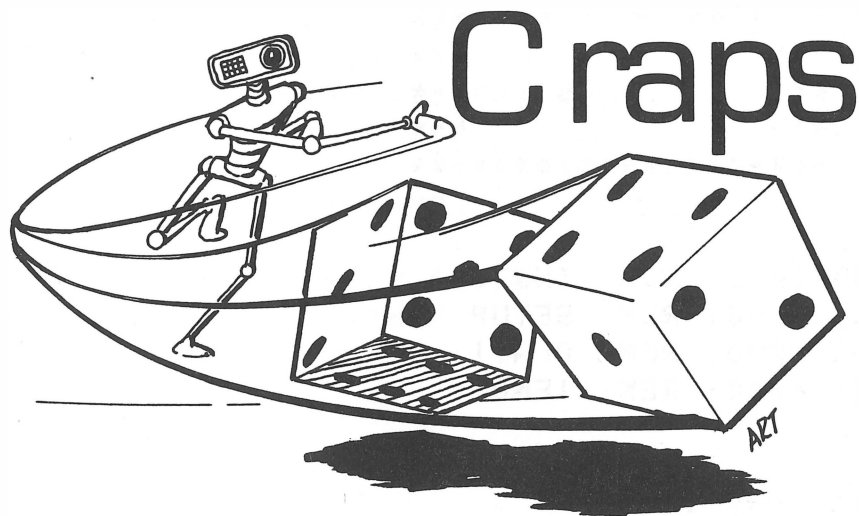
```

3020 BX = INT ( RND (1) * 20) +
      10:BY = 21:BD = 1:BF = 0:BA =
      INT ( RND (1) * 4) + 2:PB =
      0
3030 GOSUB 2300: REM BALL
3040 GOSUB 2200: REM PADDLE
3050 GOTO 3030
3060 POKE 0,250: POKE 1,100: CALL
      768: NEXT : RETURN
4000 :
4001 REM *** END
4002 :
4010 HOME : PRINT "THE GAME IS O
      VER !!!"
4020 PRINT "YOUR SCORE IS : "SC"
      (" ;
4030 IF SC < 100 THEN PRINT "LO
      USY)": RETURN
4031 IF SC < 200 THEN PRINT "PO
      OR)": RETURN
4032 IF SC < 300 THEN PRINT "SO
      -SO)": RETURN
4033 IF SC < 400 THEN PRINT "BL
      AH)": RETURN
4034 IF SC < 500 THEN PRINT "GO
      OD)": RETURN
4035 IF SC < 600 THEN PRINT "GR
      EAT)": RETURN
4036 IF SC < 700 THEN PRINT "EX
      CELLENT)": RETURN
4037 IF SC < 720 THEN PRINT "FA
      NTASTIC " : RETURN
4038 PRINT "PERFECT!!! " : RETURN

```







Craps is a simplified version of the popular dice game. You are given a \$1500 stake to play with until the money is gone. To stop playing, bet 0 dollars. Let's look into the program. Type: LIST 2190. To better understand the function of any one line, delete the line by typing in the line number, return, and run. List line 2200. D1 and D2 are the variables for the dice. Once you understand line 2200, you will be able to "fix" the dice. As is, the outcome is random. But by changing this line, you can control their total. Line 2011 changes the color to white (COLOR= 15), and lines 2022 and 2023 draw the perimeter. Change 2011 so that it reads: 2011 COLOR= 1. Now when you run the program, the perimeter will be magenta (COLOR= 1). When the dice are rolled, a graphic representation (drawing) of the random number (between one and six) is displayed on the screen. There are two cubes (dies) which, when added together, comprise the total. Line 2330 tells the computer to branch to one of the six given lines, depending on the RaNDom value of DD. The six lines (2331-2336) draw a configuration of a die, equal to 1,2,3,4,5, or 6.

```

10 REM *****
11 REM *** ***
12 REM *** CRAPS ***
13 REM *** ***
14 REM *****
15 REM
16 REM
20 GOSUB 1000: REM INSTS
30 GOSUB 2000: REM SETUP
40 GOSUB 3000: REM PLAY!
50 GOSUB 4000: REM !END!
60 END
1000 :
1001 REM *** INSTS
1002 :
1010 TEXT : NORMAL : HOME
1020 VTAB 3: HTAB 13: PRINT "***
      CRAPS ***"
1030 VTAB 6
1031 PRINT "THIS IS A DICE GAME
      CALLED **CRAPS** ."
1035 PRINT
1040 PRINT "TO PLAY, YOU WAGER A
      PORTION OF YOUR MONEY ON
      A ROLL OF THE DICE. HERE A
      RE THE RULES...."
1045 PRINT
1050 PRINT "YOU WIN IF THE FIRST
      ROLL IS EITHER 7 OR 11. CON
      VERSELY, YOU LOSE IF THE FIR
      ST ROLL IS 2, 3, OR 12."
1060 PRINT
1065 PRINT "IF YOU GET A 4,5,6,8
      ,9, OR 10 ON YOUR FIRST RO
      LL, IT IS REFERRED TO AS YOU
      R *POINT*. YOU MUST CONTI
      NUE ROLLING "
1070 PRINT "UNTIL YOU: 1) ROLL A
      7, WHICH IS CALLED *CRAPPIN
      G OUT* WHEREBY YOU LOSE, OR
      2) YOU ROLL A NUMBER EQUAL
      TO YOUR POINT...YOU WIN!"
1075 PRINT

```



```

1078 INPUT "PRESS RETURN WHEN RE
      ADY TO CONTINUE";ANS$
1080 HOME : VTAB 3: HTAB 13: PRINT
      "*** CRAPS ***": VTAB 7
1100 VTAB 12
1110 PRINT "TO QUIT THE GAME, BE
      T 0 DOLLARS."
1990 VTAB 23: INPUT "PRESS RETUR
      N WHEN READY TO CONTINUE : "
      ;ANS$
1995 RETURN
2000 :
2001 REM *** SETUP
2002 :
2010 GR
2011 COLOR= 15
2020 MNY = 1500
2022 HLIN 0,39 AT 00: HLIN 0,39 AT
      39
2023 VLIN 0,39 AT 00: VLIN 0,39 AT
      39
2190 RETURN
2200 D1 = INT ( RND (1) * 6) + 1
      :D2 = INT ( RND (1) * 6) +
      1
2210 GOSUB 2300: GOSUB 2310
2220 IF PEEK ( - 16384) < 128 THEN
      2200
2230 POKE - 16368,0: RETURN
2300 DD = D1:DX = 10: GOTO 2320
2310 DD = D2:DX = 24: GOTO 2320
2320 COLOR= INT ( RND (1) * 14)
      + 1
2325 FOR I = 20 TO 26: HLIN DX,D
      X + 6 AT I: NEXT
2330 COLOR= 15: ON DD GOTO 2331,
      2332,2333,2334,2335,2336
2331 PLOT DX + 3,23: GOTO 2340
2332 PLOT DX + 1,21: PLOT DX + 5
      ,25: GOTO 2340
2333 PLOT DX + 1,21: PLOT DX + 3
      ,23: PLOT DX + 5,25: GOTO 23
      40

```

```

2334  PLOT DX + 1,21: PLOT DX + 5
      ,21: PLOT DX + 1,25: PLOT DX
      + 5,25: GOTO 2340
2335  PLOT DX + 1,21: PLOT DX + 5
      ,21: PLOT DX + 1,25: PLOT DX
      + 5,25: PLOT DX + 3,23: GOTO
      2340
2336  PLOT DX + 1,21: PLOT DX + 5
      ,21: PLOT DX + 1,25: PLOT DX
      + 5,25: PLOT DX + 1,23: PLOT
      DX + 5,23: GOTO 2340
2340  FOR I = 1 TO 5:XX = PEEK
      - 16336): NEXT : RETURN
3000
3001  REM *** PLAY
3002
3010  HOME : PRINT "YOU HAVE "MNY
      " DOLLARS"
3020  INPUT "HOW MUCH WILL YOU RI
      SK ON THIS BET? ";ANS$
3021  ANS = VAL (ANS$)
3022  IF ANS < 0 OR ANS > MNY OR
      ANS < > INT (ANS) THEN 301
      0
3023  IF ANS = 0 THEN RETURN
3025  HOME : PRINT "BET: "ANS"
      (ROLLING ...)"
3030  PRINT "<PRESS ANY KEY TO ST
      OP THE ROLL>"
3040  GOSUB 2200
3050  HOME : PRINT "ROLLED: "D1 +
      D2
3055  IF D1 + D2 = 2 OR D1 + D2 =
      3 OR D1 + D2 = 12 THEN PRINT
      "YOU CRAPPED OUT ...": FOR P
      A = 1 TO 1500: NEXT PA: GOTO
      3100
3060  IF D1 + D2 = 7 OR D1 + D2 =
      11 THEN PRINT "YOU WON THAT
      TOSS ...": FOR PA = 1 TO 15
      00: NEXT PA: GOTO 3200
3061  PRINT "YOUR POINT IS "D1 +
      D2:PT = D1 + D2: FOR PA = 1 TO
      1200: NEXT PA

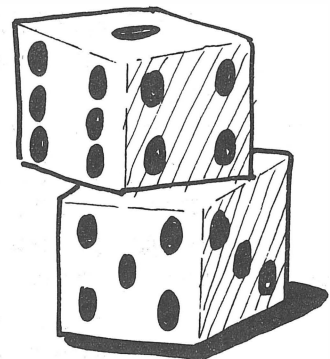
```

```

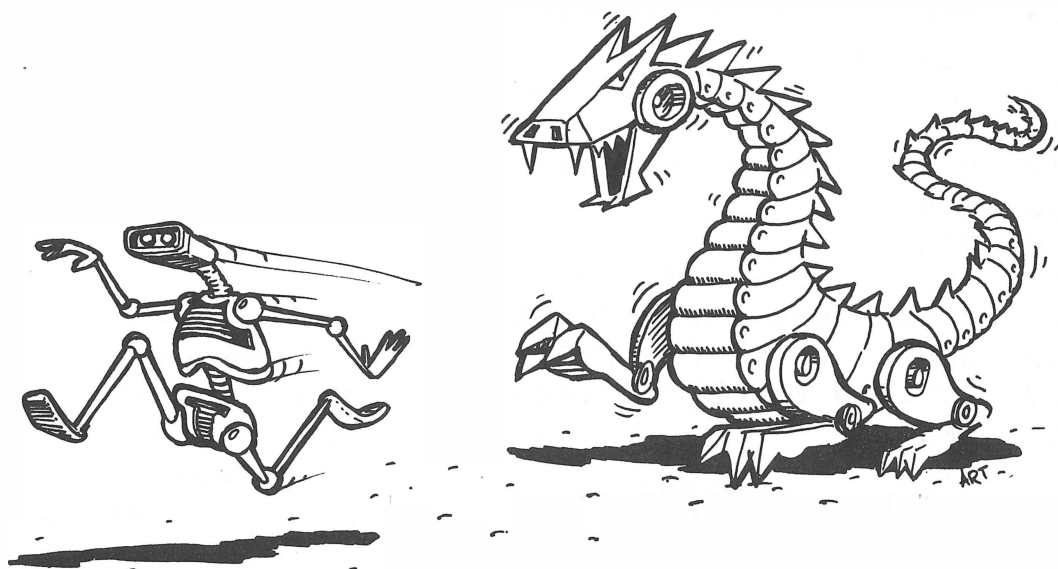
3065 HOME : PRINT "BET: "ANS" PT
      : "PT" (ROLLING AGAIN,...)"
3066 PRINT "<PRESS ANY KEY TO ST
      OP THE ROLL>": GOSUB 2200
3070 HOME : PRINT "ROLLED: "D1 +
      D2
3080 IF D1 + D2 = 7 THEN PRINT
      "YOU CRAPPED OUT ...": FOR P
      A = 1 TO 1500: NEXT PA: GOTO
      3100
3085 IF D1 + D2 = PT THEN PRINT
      "YOU GOT YOUR POINT !!!": FOR
      PA = 1 TO 1500: NEXT PA: GOTO
      3200
3090 PRINT "YOU MUST ROLL AGAIN,
      ..": FOR PA = 1 TO 1200: NEXT
      PA: GOTO 3065
3100 MNY = MNY - ANS: IF MNY = 0 THEN
      RETURN
3110 GOTO 3010
3200 MNY = MNY + ANS: GOTO 3010
4000 :
4001 REM *** END
4002 :
4010 FOR I = 0 TO 39: COLOR= INT
      ( RND (1) * 15) + 1: HLIN 0,
      39 AT I: HLIN 0,39 AT 39 - I
      : VLIN 0,39 AT I: VLIN 0,39 AT
      39 - I:XX = PEEK ( - 16336)
      - PEEK ( - 16336) - PEEK
      ( - 16336): NEXT
4020 FOR PA = 1 TO 1500: NEXT PA

4030 HOME : PRINT "YOU STOPPED W
      ITH "MNY" DOLLARS"
4040 IF MNY < 100 THEN PRINT "B
      ETTER STICK TO THE SLOT MACH
      INES!": RETURN
4041 IF MNY < 500 THEN PRINT "N
      OT SO GOOD": RETURN
4042 IF MNY < 1500 THEN PRINT "
      NOT BAD AT ALL....": RETURN
4043 PRINT "THAT'S GREAT! ";
4044 IF MNY > 6000 THEN PRINT "
      WOW, WHATTA ROLL!!!"
4045 RETURN

```







# Dragon's Lair

In this game the maze is constructed before your eyes but becomes invisible when completed. You must avoid the man-eating dragon and escape the maze. Otherwise you will attend the dragon's dinner—as his main course! Only a good memory will save you from this terrible fate. To experiment with the graphics, type: LIST 2025. Again, change any line whose function is unclear. Specifically, begin by looking at line 2020. Change COLOR = 15 to COLOR = 4. Lines 3030-3033 check to see which direction you want to move. Each of the four lines which the computer might branch to, either increments or decrements the X or Y coordinate. For instance, line 3100 handles the job when the user has entered K (to move right). When you try to move right, the Y coordinate is unchanged. The X coordinate is increased ( $X2 = PX + 1$ ). At the origin the XY coordinates are 0,0. As you move to the right, the Y coordinate stays at 0, but the X coordinate continually increases. Beginning at the origin, 0,0, the coordinates would look like this:

0,1 0,2 0,3 0,4.....0,38 0,39.

```

10 REM *****
11 REM *** ***
12 REM *** DRAGON'S LAIR ***
13 REM *** ***
14 REM *****
15 REM
16 REM
20 GOSUB 1000: REM INSTS
30 GOSUB 2000: REM SETUP
40 GOSUB 3000: REM PLAY!
50 GOSUB 4000: REM !END!
60 END
1000 :
1001 REM *** INSTS
1002 :
1010 TEXT : NORMAL : HOME
1020 VTAB 3: HTAB 10: PRINT "***
    DRAGONS LAIR ***"
1030 VTAB 7: PRINT "YOU WILL BE
    PLACED IN A MAZE WITH A MAN-
    EATING DRAGON."
1040 PRINT : PRINT "YOUR PROBLEM
    IS TO TRY AND ESCAPE FROM
    THE MAZE BEFORE THE DRAGON M
    AKES A MEAL OUT OF YOU."
1050 PRINT : PRINT "YOU MOVE BY
    PRESSING : ": PRINT : PRINT
    "      UP -      'I' ": PRINT
    "      DOWN -      'M' ": PRINT
    "      RIGHT -      'K' ": PRINT
    "      LEFT -      'J' "
1060 VTAB 23: INPUT "HIT RETURN
    WHEN READY TO CONTINUE : ";A
    NS$
1070 HOME : VTAB 3: HTAB 10: PRINT
    "*** DRAGONS LAIR ***"
1080 VTAB 7: PRINT "YOU MAY PLAY
    IN ONE OF THE FOLLOWING TWO
    MODES: ": PRINT : PRINT "  N
    ORMAL": PRINT "WHERE THE DRA
    GON MOVES ONLY AFTER YOU  M
    AKE A MOVE ... "

```

```

1090 PRINT : PRINT "    REAL-TIME
    ": PRINT "WHERE THE DRAGON'S
    PURSUIT IS CONSTANT, WHETHE
    R YOU MOVE OR NOT ! "
1100 PRINT : PRINT "    WILL YOU
    PLAY:" : PRINT : PRINT "
    N)ORMAL          -OR-          R)EAL
    -TIME"
1110 VTAB 23: CALL - 958: INPUT
    "WHAT IS YOUR CHOICE (N/R)";
    ANS$:ANS$ = LEFT$(ANS$,1):
    IF ANS$ < > "N" AND ANS$ <
    > "R" THEN 1110
1990 RETURN
2000 :
2001 REM *** SETUP
2002 :
2010 DIM MA(13,13,4)
2015 DEF FN R(X) = INT ( RND (
    1) * X) + 1
2020 RX = FN R(13):RY = FN R(13
    ): GR : COLOR= 15: FOR I = 0
    TO 18 STEP 3: HLINE 0,39 AT
    I: HLINE 0,39 AT 39 - I: VLINE
    0,39 AT I: VLINE 0,39 AT 39 -
    I: NEXT
2021 HOME : VTAB 22: PRINT "    <
    << I'M NOW BUILDING THE LAIR
    >>>"
2025 COLOR= 0: CNT = 1: GOTO 2070

2035 IF FN R(10) = 1 THEN 2100
2070 ON FN R(4) GOTO 2075,2080,
    2085,2090
2075 IF RX = 13 THEN 2035
2076 IF MA(RX,RY,1) THEN 2035
2077 IF MA(RX + 1,RY,0) THEN 203
    5
2078 VLINE 3 * RY - 2,3 * RY - 1 AT
    3 * RX:MA(RX,RY,0) = MA(RX,R
    Y,0) + 1:MA(RX,RY,1) = 1:RX =
    RX + 1:MA(RX,RY,3) = 1:MA(RX
    ,RY,0) = MA(RX,RY,0) + 1

```

```

2079 GOTO 2095
2080 IF RY = 13 THEN 2035
2081 IF MA(RX,RY,2) THEN 2035
2082 IF MA(RX,RY + 1,0) THEN 203
5
2083 HLIN 3 * RX - 2,3 * RX - 1 AT
3 * RY:MA(RX,RY,0) = MA(RX,R
Y,0) + 1:MA(RX,RY,2) = 1:RY =
RY + 1:MA(RX,RY,4) = 1:MA(RX
,RY,0) = MA(RX,RY,0) + 1
2084 GOTO 2095
2085 IF RX = 1 THEN 2035
2086 IF MA(RX,RY,3) THEN 2035
2087 IF MA(RX - 1,RY,0) THEN 203
5
2088 VLIN 3 * RY - 2,3 * RY - 1 AT
3 * (RX - 1):MA(RX,RY,0) = M
A(RX,RY,0) + 1:MA(RX,RY,3) =
1:RX = RX - 1:MA(RX,RY,1) =
1:MA(RX,RY,0) = MA(RX,RY,0) +
1
2089 GOTO 2095
2090 IF RY = 1 THEN 2035
2091 IF MA(RX,RY,4) THEN 2035
2092 IF MA(RX,RY - 1,0) THEN 203
5
2093 HLIN 3 * RX - 2,3 * RX - 1 AT
3 * (RY - 1):MA(RX,RY,0) = M
A(RX,RY,0) + 1:MA(RX,RY,4) =
1:RY = RY - 1:MA(RX,RY,2) =
1:MA(RX,RY,0) = MA(RX,RY,0) +
1
2094 GOTO 2095
2095 IF MA(RX,RY,0) = 1 THEN CNT
= CNT + 1: IF CNT = 169 THEN
2200
2097 GOTO 2035
2100 RX = FN R(13):RY = FN R(13
): IF MA(RX,RY,0) = 0 OR MA(
RX,RY,0) = 4 THEN 2100
2105 GOTO 2035
2200 GR : COLOR= 15: VLIN 0,39 AT
39: HLIN 0,39 AT 39: VLIN 0,
39 AT 0: HLIN 0,39 AT 0

```



```

2210 PX = 1:PY = FN R(13): COLOR=
      8:XX = PX:YY = PY: GOSUB 250
      0
2215 WY = FN R(13): COLOR= 0: VLIN
      WY * 3 - 2,WY * 3 - 1 AT 39
2220 MX = 13:MY = WY: COLOR= 1:XX
      = MX:YY = MY: GOSUB 2500
2300 POKE 768,173: POKE 769,48: POKE
      770,192: POKE 771,136: POKE
      772,208: POKE 773,4: POKE 77
      4,198: POKE 775,1: POKE 776,
      240
2310 POKE 777,8: POKE 778,202: POKE
      779,208: POKE 780,246: POKE
      781,166: POKE 782,0: POKE 78
      3,76: POKE 784,0: POKE 785,3
      : POKE 786,96
2320 FOR I = 1 TO 13: FOR J = 1 TO
      13:MA(I,J,0) = 0: NEXT J,I
2400 HOME : PRINT "K) FOR RIGHT
      I) FOR UP": PRINT "M) F
      OR DOWN      J) FOR LEFT"
2490 RETURN
2500 REM *** DRAW A SQUARE
2505 FOR I = XX * 3 - 2 TO XX *
      3 - 1: FOR J = YY * 3 - 2 TO
      YY * 3 - 1: PLOT I,J: NEXT J
      ,I: RETURN
3000 :
3001 REM *** PLAY
3002 :
3010 IF PEEK ( - 16384) < 128 AND
      ANS$ = "R" THEN 3700
3015 IF PEEK ( - 16384) < 128 THEN
      3015
3020 KEY = PEEK ( - 16384): POKE
      - 16368,0
3025 KEY$ = CHR$ (KEY - 128)
3030 IF KEY$ = "K" THEN 3100
3031 IF KEY$ = "M" THEN 3200
3032 IF KEY$ = "J" THEN 3300
3033 IF KEY$ = "I" THEN 3400
3040 POKE 0,200: POKE 1,100: CALL
      768: GOTO 3700

```

```

3100 IF MA(PX,PY,1) THEN X2 = PX
      + 1:Y2 = PY: GOTO 3500
3110 POKE 0,20: POKE 1,20: CALL
      768: COLOR= 4: VLIN PY * 3 -
      3,PY * 3 AT PX * 3: GOTO 370
      0
3200 IF MA(PX,PY,2) THEN X2 = PX
      :Y2 = PY + 1: GOTO 3500
3210 POKE 0,20: POKE 1,20: CALL
      768: COLOR= 4: HLIN PX * 3 -
      3,PX * 3 AT PY * 3: GOTO 370
      0
3300 IF MA(PX,PY,3) THEN X2 = PX
      - 1:Y2 = PY: GOTO 3500
3310 POKE 0,20: POKE 1,20: CALL
      768: COLOR= 4: VLIN PY * 3 -
      3,PY * 3 AT (PX - 1) * 3: GOTO
      3700
3400 IF MA(PX,PY,4) THEN X2 = PX
      :Y2 = PY - 1: GOTO 3500
3410 POKE 0,20: POKE 1,20: CALL
      768: COLOR= 4: HLIN PX * 3 -
      - 3,PX * 3 AT (PY - 1) * 3: GOTO
      3700
3500 COLOR= 0:XX = PX:YY = PY: GOSUB
      2500: COLOR= 8:XX = X2:YY =
      Y2: GOSUB 2500:PX = X2:PY =
      Y2: GOTO 3700
3700 IF PX = 13 AND PY = WY THEN
      WIN = 1: RETURN
3705 IF PX = MX AND PY = MY THEN
      XX = MX:YY = MY: COLOR= 1: GOSUB
      2500:WIN = 0: RETURN
3709 ON FN R(4) GOTO 3710,3711,
      3712,3713
3710 IF MX < PX THEN 3720
3711 IF MY < PY THEN 3730
3712 IF MX > PX THEN 3740
3713 IF MY > PY THEN 3750
3714 GOTO 3710
3720 IF MX = 13 THEN 3730
3722 IF MA(MX,MY,0) > 5 THEN 372

```

6

```

3724 IF NOT MA(MX,MY,1) THEN 37
30
3726 X2 = MX + 1:Y2 = MY:XX = MX:
YY = MY: COLOR= 0: GOSUB 250
0:XX = X2:YY = Y2: COLOR= 1:
GOSUB 2500:MX = X2:MY = Y2:
MA(MX,MY,0) = MA(MX,MY,0) +
1: GOTO 3800
3730 IF MY = 13 THEN 3740
3732 IF MA(MX,MY,0) > 5 THEN 373
6
3734 IF NOT MA(MX,MY,2) THEN 37
40
3736 X2 = MX:Y2 = MY + 1:XX = MX:
YY = MY: COLOR= 0: GOSUB 250
0:XX = X2:YY = Y2: COLOR= 1:
GOSUB 2500:MX = X2:MY = Y2:
MA(MX,MY,0) = MA(MX,MY,0) +
1: GOTO 3800
3740 IF MX = 1 THEN 3750
3742 IF MA(MX,MY,0) > 5 THEN 374
6
3744 IF NOT MA(MX,MY,3) THEN 37
50
3746 X2 = MX - 1:Y2 = MY:XX = MX:
YY = MY: COLOR= 0: GOSUB 250
0:XX = X2:YY = Y2: COLOR= 1:
GOSUB 2500:MX = X2:MY = Y2:
MA(MX,MY,0) = MA(MX,MY,0) +
1: GOTO 3800
3750 IF MY = 1 THEN 3720
3752 IF MA(MX,MY,0) > 5 THEN 375
6
3754 IF NOT MA(MX,MY,4) THEN 37
20
3756 X2 = MX:Y2 = MY - 1:XX = MX:
YY = MY: COLOR= 0: GOSUB 250
0:XX = X2:YY = Y2: COLOR= 1:
GOSUB 2500:MX = X2:MY = Y2:
MA(MX,MY,0) = MA(MX,MY,0) +
1: GOTO 3800
3800 IF PX = 13 AND PY = WY THEN
WIN = 1: RETURN

```

```

3805  IF PX = MX AND PY = MY THEN
      WIN = 0: RETURN
3810  GOTO 3010
3990  RETURN
4000  :
4001  REM  *** END
4002  :
4010  IF (WIN) THEN 4100
4020  HOME : FOR I = 10 TO 50: POKE
      0,I: POKE 1,20: CALL 768: NEXT
      : FOR I = 1 TO 3: POKE 0,200
      : POKE 1,150: CALL 768: NEXT
      : POKE 0,240: POKE 1,250: CALL
      768
4035  VTAB 22: PRINT "SORRY, BUT
      THE DRAGON GOT YOU...."
4095  RETURN
4100  HOME : FOR I = 50 TO 20 STEP
      - 1: POKE 0,I: POKE 1,20: CALL
      768: POKE 0,I - 1: POKE 1,20
      : CALL 768: NEXT
4110  VTAB 22: PRINT "***** YOU W
      ON ***** (BUT NOW THE DRAGO
      NIS EVEN HUNGRIER!)"
4990  RETURN

```

In this popular word game, you try to surmise the 'secret word' by guessing individual letters contained therein. Failure to divine the complete word will result in the completed figure being hanged. Because the graphics are quite clear in this program, a few lines will be highlighted. Type: LIST-2140. Line 2100 draws a white border around the gallows. Line 2110 draws the gallows. Line 2120 completes what line 2110 started. Lines 2130-2135 draw the stairs leading up to the hanging platform. Each time you make an incorrect guess, line 3070 instructs the program to perform a subroutine. These subroutines each draw a separate part of the prisoner's body. Each line (2200-2900) draws a piece of the man. Line 2200 draws the head, line 2300 draws the eyes, line 2400 draws the mouth, and so on. You are encouraged to experiment with these lines and to change or omit any line whose function is unclear.

```

10 REM *****
11 REM *** ***
12 REM *** HANG MAN ***
13 REM *** ***
14 REM *****
15 REM
16 REM
20 GOSUB 1000: REM INSTS
30 GOSUB 2000: REM SETUP
40 GOSUB 3000: REM PLAY!
50 GOSUB 4000: REM !END!
60 END

1000 :
1001 REM *** INSTS
1002 :
1010 TEXT : NORMAL : HOME
1020 VTAB 3: HTAB 12: PRINT "***
      HANG MAN ***"
1025 VTAB 7
1030 PRINT "IN THIS GAME YOU ARE
      GIVEN THE FORMAT OF A WORD.
      YOU TRY TO SPELL OUT THE
      MYSTERY WORD BY GUESSING
      ONE LETTER AT A TIME. "
1035 PRINT
1040 PRINT "IF THE LETTER IS IN
      THE WORD, I WILL EXPOSE A
      LL OCCURRENCES OF THAT LETTE
      R, IN THEIR CORRECT POSITIO
      N WITHIN THE WORD. IF YO
      U GUESS A LETTER NOT FOUND
      IN THE WORD, THEN I WILL ADD
      A PART TO THE MAN IN THE GA
      LLOWS.
1045 PRINT
1050 PRINT "WHEN THE MAN IS COMP
      LETE, (HEAD, MOUTH, EYES, AR
      MS, AND LEGS), HE IS HUNG A
      ND YOU LOSE. TO WIN, GUESS
      THE ENTIRE WORD CORRECTLY."
1055 PRINT
1060 INPUT "HIT RETURN WHEN READ
      Y TO CONTINUE : ";ANS$

```

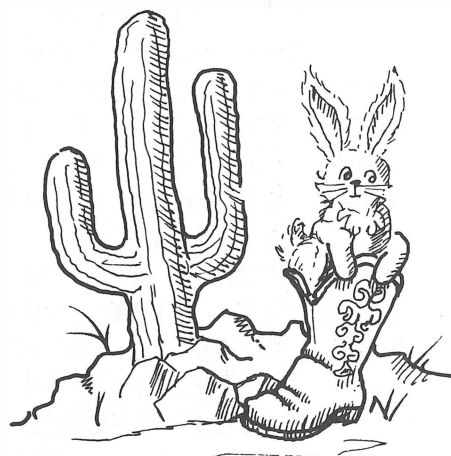
```

1990 RETURN
2000 :
2001 REM *** SETUP
2002 :
2010 RESTORE
2020 FOR I = 1 TO INT ( RND (1)
      * 30) + 1: READ WRD$: NEXT

2030 DATA "PENCIL","COMPUTER","P
RINTER","ELEPHANT","NOTEBOOK
"
2031 DATA "HANGMAN","POSTER","CE
ILING","FOOTBALL","EVERGREEN
"
2032 DATA "YESTERDAY","MIRROR","
PICTURE","CARPET","MONOPOLY"
2033 DATA "SCOUNDREL","PROFILE",
"EQUIPMENT","FOUNTAIN","LAVI
SH"
2034 DATA "COOKIES","PLEASURE","
ROUTINE","TEACHER","REGULAR"

2035 DATA "BARBECUE","BARRIER","
PAVEMENT","THOUGHTFUL","MARR
IAGE"
2050 DIM GU$(15):WL = LEN (WRD$
): FOR I = 1 TO WL:GU$(I) =
CHR$ (95): NEXT
2055 GUESSED$ = ""
2100 GR : COLOR= 15: HLIN 0,39 AT
0: HLIN 0,39 AT 39: VLIN 0,3
9 AT 39: VLIN 0,39 AT 0
2110 HLIN 5,34 AT 28: VLIN 7,35 AT
5: VLIN 29,35 AT 34: HLIN 6,
21 AT 7
2120 PLOT 6,11: PLOT 7,10: PLOT
8,9: PLOT 9,8: PLOT 20,8: PLOT
20,9
2130 COLOR= 8: HLIN 19,21 AT 29:
HLIN 17,23 AT 31: HLIN 15,2
5 AT 33: HLIN 13,27 AT 35
2135 COLOR= 9: HLIN 18,22 AT 30:
HLIN 16,24 AT 32: HLIN 14,2
6 AT 34

```



```

2190 RETURN
2200 COLOR= 2: HLIN 19,21 AT 10:
      HLIN 19,21 AT 16: VLIN 12,1
      4 AT 17: VLIN 12,14 AT 23: PLOT
      18,11: PLOT 22,11: PLOT 18,1
      5: PLOT 22,15: RETURN
2300 COLOR= 7: PLOT 18,12: PLOT
      19,12: PLOT 21,12: PLOT 22,1
      2: PLOT 19,13: PLOT 21,13: RETURN

2400 COLOR= 6: PLOT 19,15: PLOT
      20,14: PLOT 21,15: RETURN
2500 COLOR= 3: VLIN 17,22 AT 20:
      RETURN
2600 COLOR= 12: HLIN 17,19 AT 19
      : VLIN 19,21 AT 17: RETURN
2700 COLOR= 12: HLIN 21,23 AT 19
      : VLIN 19,21 AT 23: RETURN
2800 COLOR= 4: PLOT 19,23: PLOT
      18,24: HLIN 16,18 AT 25: RETURN

2900 COLOR= 4: PLOT 21,23: PLOT
      22,24: HLIN 22,24 AT 25: RETURN

3000
3001 REM *** PLAY
3002
3010 HOME : PRINT "WORD: "; FOR
      I = 1 TO WL: PRINT GU$(I);: NEXT
      : PRINT
      FOR I = 1 TO WL: IF GU$(I) <
        > CHR$(95) THEN NEXT :WO
      L = 1: RETURN
3020 PRINT "GUESSES: ";GU$
3030 PRINT : PRINT "WHAT IS YOUR
      GUESS ==> ";: GET ANS$
3040 IF ANS$ < "A" OR ANS$ > "Z"
      THEN 3010
3045 FOR I = 1 TO WL
3050 IF MID$(GU$,I,1) = ANS$ THEN
      VTAB 23: HTAB 1: CALL - 95
      8: PRINT "<<< THAT'S ALREADY
      BEEN GUESSED >>>": FOR PA =
      1 TO 1000: NEXT PA: GOTO 301
      0

```



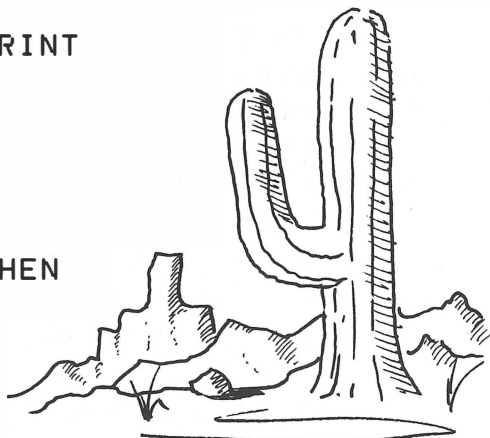
```

3055 NEXT
3060 RC = 0: FOR I = 1 TO WL: IF
    MID$ (WR$,I,1) = ANS$ THEN
    GU$(I) = ANS$:RC = RC + 1
3065 NEXT : IF RC > 0 THEN 3010
3070 GU$ = GU$ + ANS$: ON LEN (G
    U$) GOSUB 2200,2300,2400,250
    0,2600,2700,2800,2900
3075 XX = PEEK ( - 16336) + PEEK
    ( - 16336) + PEEK ( - 16336
    ) + PEEK ( - 16336)
3080 IF LEN (GU$) < 8 THEN 3010

3090 WOL = 0: RETURN
4000 :
4001 REM *** END
4002 :
4010 FOR I = 0 TO 39: COLOR= INT
    ( RND (1) * 15) + 1: VLIN 0,
    39 AT I:XX = PEEK ( - 16336
    ) - PEEK ( - 16336) - XX =
    PEEK ( - 16336) - XX = PEEK
    ( - 16336): NEXT
4020 FOR PA = 1 TO 1500: NEXT PA

4030 TEXT : HOME : VTAB 3
4040 PRINT "THE GAME IS OVER !!!
"
4050 IF WOL THEN PRINT : PRINT
    : PRINT "YOU GUESSED THE COR
    RECT WORD, AND THE": PRINT "
    PRISONER WILL GO FREE !!!"
4055 IF NOT WOL THEN PRINT : PRINT
    : PRINT "THE PRISONER HAS BE
    EN HUNG, AS YOU": PRINT "FAI
    LED TO GUESS: "WRD$" !!!"
4060 PRINT : PRINT INPUT "DO YO
    U WISH TO PLAY AGAIN? ";ANS$
    : IF LEFT$ (ANS$,1) = "Y" THEN
    RUN
4990 RETURN

```





In this game you build a maze for the maze-loving Itche worm. Upon instruction, the worm will attempt to solve the maze. Though not a game per se, it is an ingenious utilization of graphics. The two routines which control color begin at 2000 and 2700. Line 2210 sets the color of the maze perimeter and then branches to 2700. Whether you are moving, plotting, or erasing, you must specify which direction you intend to travel. Lines 2715-2740 check to see which of the four directions you chose and will cause the program to branch to the accommodating line (2750-2780). Lines to experiment with include 2010, 2210, and 3010.

```

10 REM *****
11 REM ***          ***
12 REM ***      ITCHE      ***
13 REM ***          ***
14 REM *****
15 REM
16 REM
20 GOSUB 1000: REM  INSTS
30 GOSUB 2000: REM  SETUP
40 GOSUB 3000: REM  PLAY!
50 GOSUB 4000: REM  !END!
60 END
1000 :
1001 REM  *** INSTS
1002 :
1010 TEXT : NORMAL : HOME
1020 VTAB 2: HTAB 13: PRINT "***
      ITCHE ***"
1030 VTAB 4: PRINT "THIS IS THE
      GAME OF ITCHE.  THE ITCHE IS
      A SPECIAL WORM WHO LIKES TO
      SOLVE MAZES. YOU ARE TO CREAT
      E A MAZE FOR THE ITCHE  TO S
      OLVE."
1040 PRINT : PRINT "BY USING THE
      FOLLOWING INSTRUCTIONS, YOU
      WILL BUILD THE MAZE WALLS.
      UPON          COMMAND, (G), IT
      CHE WILL WIND HIS WAY  THRO
      UGH THE MAZE."
1050 PRINT : PRINT "G)O    TO HAV
      E THE ITCHE WORM FIND THE
      EXIT TO YOUR MAZE."
1051 PRINT "X)IT    TO EXIT THE G
      AME."
1052 PRINT "C)LEAR   TO START TH
      E MAZE OVER."
1053 PRINT : PRINT "E)    FOR ERA
      SE:  FOLLOWED BY THESE..."
1054 PRINT "M)    FOR MOVE
      U=UP"
1055 PRINT "P)    FOR PLOT
      R=RIGHT    L=LEFT"

```

```

1056 PRINT "
      D=DOWN"
1058 PRINT
1060 INPUT "PRESS RETURN WHEN RE
      ADY TO CONTINUE : ";ANS$
1065 RETURN
2000 :
2001 REM *** SETUP
2002 :
2010 GR : HOME :CL = 1: COLOR= 4
      : HLIN 0,39 AT 0: HLIN 0,39 AT
      39: VLIN 0,39 AT 0: VLIN 0,3
      9 AT 39: COLOR= 0: PLOT 0,1:
      PLOT 39,1
2012 IX = 2:IY = 1: COLOR= 7: PLOT
      IX,IY:CLR = 16:CLR2 = 0
2015 POKE 768,173: POKE 769,48: POKE
      770,192: POKE 771,136: POKE
      772,208: POKE 773,4: POKE 77
      4,198: POKE 775,1: POKE 776,
      240
2017 POKE 777,8: POKE 778,202: POKE
      779,208: POKE 780,246: POKE
      781,166: POKE 782,0: POKE 78
      3,76: POKE 784,0: POKE 785,3
      : POKE 786,96
2019 HOME : PRINT "G)O      C)LEA
      R      X)IT      U)P      D)OWN P
      )LOT      E)RASE      M)OVE ===
      >      L)EFT      R)IGHT"
2020 GET ANS$
2030 IF ANS$ = "G" THEN 2100
2035 IF ANS$ = "P" THEN 2200
2040 IF ANS$ = "M" THEN 2300
2045 IF ANS$ = "E" THEN 2400
2050 IF ANS$ = "C" THEN 2500
2055 IF ANS$ = "X" THEN 2600
2056 IF ANS$ = CHR$ (3) THEN 26
      00
2060 POKE 0,10: POKE 1,5: CALL 7
      68: GOTO 2020
2100 REM *** GO ROUTINE
2110 RETURN

```

```

2200 REM *** PLOT ROUTINE
2210 CLR = 9: GOTO 2700
2300 REM *** MOVE ROUTINE
2310 CLR = 16: GOTO 2700
2400 REM *** ERASE ROUTINE
2410 CLR = 0: GOTO 2700
2500 REM *** CLEAR ROUTINE
2510 GOTO 2010
2600 REM *** EXIT ROUTINE
2610 HOME : PRINT : INPUT "Do yo
u wish to quit : ";ANS$: IF
LEFT$(ANS$,1) = "Y" THEN POP
: GOTO 50
2620 HOME : GOTO 2019
2700 REM *** DIRECTIONS
2710 GET ANS$:X2 = IX:Y2 = IY
2715 IF ANS$ = "R" THEN 2750
2720 IF ANS$ = "D" THEN 2760
2730 IF ANS$ = "L" THEN 2770
2740 IF ANS$ = "U" THEN 2780
2745 GOTO 2030
2750 REM *** RIGHT
2752 X2 = IX + 1: IF X2 < 39 THEN
2790
2754 X2 = 38: POKE 0,200: POKE 1,
15: CALL 768: GOTO 2790
2760 REM *** DOWN
2762 Y2 = IY + 1: IF Y2 < 39 THEN
2790
2764 Y2 = 38: POKE 0,200: POKE 1,
15: CALL 768: GOTO 2790
2770 REM *** LEFT
2772 X2 = IX - 1: IF X2 > 0 THEN
2790
2774 X2 = 1: POKE 0,200: POKE 1,1
5: CALL 768: GOTO 2790
2780 REM *** UP
2782 Y2 = IY - 1: IF Y2 > 0 THEN
2790
2784 Y2 = 1: POKE 0,200: POKE 1,1
5: CALL 768: GOTO 2790
2790 COLOR= CLR: IF CLR = 16 THEN
COLOR= C2

```

```

2792  PLOT IX,IY:C2 =  SCRN( X2,Y
      2): COLOR= 7: PLOT X2,Y2:IX =
      X2:IY = Y2: GOTO 2700
3000  :
3001  REM  *** PLAY
3002  :
3005  HOME : PRINT : PRINT "<<< I
      TCHE IS NOW SOLVING THE MAZE
      >>>"
3010  X2 = 1:Y2 = 1: COLOR= 13: PLOT
      X2,Y2
3015  COLOR= CLR: IF CLR = 16 THEN
      COLOR= C2
3016  PLOT IX,IY
3020  X3 = 0:Y3 = 1:DIR = 2
3030  IF X2 + X3 = 0 AND Y2 + Y3 =
      1 THEN RETURN
3035  IF X2 + X3 = 39 AND Y2 + Y3
      = 1 THEN RETURN
3040  IF X2 + X3 > 0 AND X2 + X3 <
      39 AND Y2 + Y3 > 0 AND Y2 +
      Y3 < 39 AND  SCRN( X2 + X3,Y
      2 + Y3) = 0 THEN 3100
3045  DI = DI - 1: IF DI < 1 THEN
      DI = 4
3050  IF DI = 1 THEN X3 = 1:Y3 =
      0: GOTO 3030
3055  IF DI = 2 THEN X3 = 0:Y3 =
      1: GOTO 3030
3060  IF DI = 3 THEN X3 = - 1:Y3
      = 0: GOTO 3030
3065  IF DI = 4 THEN X3 = 0:Y3 =
      - 1: GOTO 3030
3100  COLOR= 0: PLOT X2,Y2: COLOR=
      13:X2 = X2 + X3:Y2 = Y2 + Y3
      : PLOT X2,Y2:CL = CL + 1:DI =
      DI + 1: IF DI > 4 THEN DI =
      1
3110  POKE 0,25: POKE 1,3: CALL 7
      68: GOTO 3050
4000  :
4001  REM  *** END
4002  :

```



```

4010  COLOR= 0: PLOT X2,Y2:X2 = X
      2 + X3:Y2 = Y2 + Y3: FOR I =
      1 TO 10: COLOR= 13: PLOT X2,
      Y2: POKE 0,20: POKE 1,20: CALL
      768: COLOR= 0: PLOT X2,Y2: POKE
      0,40: POKE 1,20: CALL 768: NEXT
      I
4020  HOME : PRINT
4025  IF LEFT$ (ANS$,1) = "Y" THEN
      PRINT "<< ITCHE SAYS YOU SP
      OILED HIS FUN !!! >>": RETURN

4030  IF X2 = 39 THEN PRINT "<<<
      ITCHE HAS SOLVED THE MAZE >
      >>": PRINT "HE DID IT IN "CL
      " CLICKS...": RETURN
4040  PRINT "<<< ITCHE CANNOT SOL
      VE YOUR MAZE >>>": PRINT "HE
      IS STUCK AT THE BEGINNING..
      .": RETURN

```





This game is similar to the video dinosaur, Pong. You have five men, as does your opponent. Using the paddles, you try to deflect the ball into your opponent's men and also protect your own. The winner is the first player to eliminate all of the other player's men. The graphics are very straightforward and easy to understand. Start by typing: LIST -2043. The setup is handled almost entirely in lines 2030-2043. Line 2031 controls the color of the court's perimeter. If you want to verify this, change the color to any number through 15. Rewrite line 2032 so that it reads: 2032 HLIN 4,32 AT 12:HLIN 6,27 AT 34. Run the program to see what changes occur. Make similar changes to line 2033, then run the program. Line 2040 sets the color of the five men. Experiment with different colors. Rewrite line 2041 so that it reads: 2041 FOR I = 1 TO 33 STEP 8. Run the program. Change line 2042 so that it reads: 2042 VLIN I,I+4 AT 8:VLIN I,I+2 AT 31. Hopefully, these visual modifications will help you to understand the function of each line. Next, we will look at how the ball is controlled, and why it is white. Line 3075 sets the color to white, (color =15). If you want to change the color of the ball, then change line 3075. Lines 3070,3071,3080,3085-6-7, and others, control the movement of the ball. Line 3090 specifically controls the ball when it hits either back wall. To confirm this, type: 3090. Hit RETURN, then rerun the program.

```

10  REM  *****
11  REM  ***                      ***
12  REM  ***   KNOCK   OUT   ***
13  REM  ***                      ***
14  REM  *****
15  REM
16  REM
20  GOSUB 1000: REM  INSTS
30  GOSUB 2000: REM  SETUP
40  GOSUB 3000: REM  PLAY!
50  GOSUB 4000: REM  !END!
60  END
1000 :
1001  REM  *** INSTS
1002 :
1010  TEXT : NORMAL : HOME
1020  VTAB 3: HTAB 11: PRINT "***
      KNOCK   OUT ***"
1030  VTAB 7: PRINT "*KNOCK OUT*
      IS A TWO PLAYER GAME THAT IS
      SIMILAR TO PONG.  EACH PLAYE
      R HAS A      PADDLE WITH WHIC
      H HE TRIES TO RETURN THEBALL
      INTO HIS OPPONENT'S COURT,"

1040  PRINT
1050  PRINT "THE OBJECT OF THE GA
      ME IS TO KO (KNOCK   OUT) YOU
      R OPPONENT'S MEN WHILE DEFEN
      DINGYOUR OWN."
1055  PRINT
1060  PRINT "THE FIRST PLAYER TO
      KNOCK OUT ALL OF THEOPPOSITI
      ON'S MEN IS THE WINNER."
1070  VTAB 22: INPUT "HIT RETURN
      WHEN READY TO CONTINUE : ";A
      NS$
1080  RETURN
2000 :
2001  REM  *** SETUP

```

```

2002 :
2010 POKE 768,173: POKE 769,48: POKE
    770,192: POKE 771,136: POKE
    772,208: POKE 773,4: POKE 77
    4,198: POKE 775,1: POKE 776,
    240
2020 POKE 777,8: POKE 778,202: POKE
    779,208: POKE 780,246: POKE
    781,166: POKE 782,0: POKE 78
    3,76: POKE 784,0: POKE 785,3
    : POKE 786,96
2030 GR : HOME
2031 COLOR= 15
2032 HLIN 0,39 AT 0: HLIN 0,39 AT
    39
2033 VLIN 0,39 AT 0: VLIN 0,39 AT
    39
2040 COLOR= 12
2041 FOR I = 6 TO 30 STEP 6
2042 VLIN I,I + 3 AT 3: VLIN I,I
    + 3 AT 36
2043 NEXT I
2050 LET O0 = 1: LET O1 = 1
2060 PRINT "PLAYER # 1
        PLAYER # 2"
2070 FOR I = 1 TO 100: GOSUB 210
    0: NEXT I
2080 LET SF = 1
2090 RETURN
2100 :
2110 REM *** PADDLE CONTROL
2120 :
2130 P0 = INT ( PDL (0) * 10 / 7
    5 + 1):P1 = INT ( PDL (1) *
    10 / 75 + 1)
2140 IF P0 < > 00 THEN COLOR=
    0: VLIN 00,00 + 3 AT 14: COLOR=
    4:00 = P0: VLIN 00,00 + 3 AT
    14
2150 IF P1 < > 01 THEN COLOR=
    0: VLIN 01,01 + 3 AT 25: COLOR=
    1:01 = P1: VLIN 01,01 + 3 AT
    25

```

```

2160 RETURN
3000 :
3001 REM *** PLAY
3002 :
3060 L1 = 5:L2 = 5
3070 BX = INT ( RND (1) * 10) +
      16:BY = INT ( RND (1) * 10)
      + 16:BD = INT ( RND (1) *
      2) * 2 - 1
3071 BS = INT ( RND (1) * 7) - 3
      : IF BS = 0 THEN 3071
3075 COLOR= 15: PLOT BX,BY
3080 X2 = BX + BD:Y2 = BY + BS: GOSUB
      2100
3085 IF SCRN( BX,BY) = 15 THEN
      3090
3086 BX = BX - BD:BY = BY - BS:BD
      = - BD:BS = - BS: COLOR=
      15: PLOT BX,BY: POKE 0,50: POKE
      1,10: CALL 768
3087 IF BY < 1 OR BY > 38 THEN B
      Y = BY + BS:BS = - BS
3089 GOTO 3080
3090 IF X2 < 1 OR X2 > 38 THEN B
      D = - BD: POKE 0,100: POKE
      1,10: CALL 768: GOTO 3080
3100 IF Y2 < 1 OR Y2 > 38 THEN B
      S = - BS: POKE 0,100: POKE
      1,10: CALL 768: GOTO 3080
3110 IF SCRN( X2,Y2) = 0 THEN 3
      150
3111 IF SCRN( X2,Y2) < > 12 THEN
      3125
3112 COLOR= 0: VLIN INT (Y2 / 6
      ) * 6, INT (Y2 / 6) * 6 + 3 AT
      X2: POKE 0,25: POKE 1,2: CALL
      768
3115 IF X2 = 3 THEN L1 = L1 - 1
3116 IF X2 = 36 THEN L2 = L2 - 1

3117 IF L1 = 0 OR L2 = 0 THEN 35
      00
3120 GOTO 3150

```

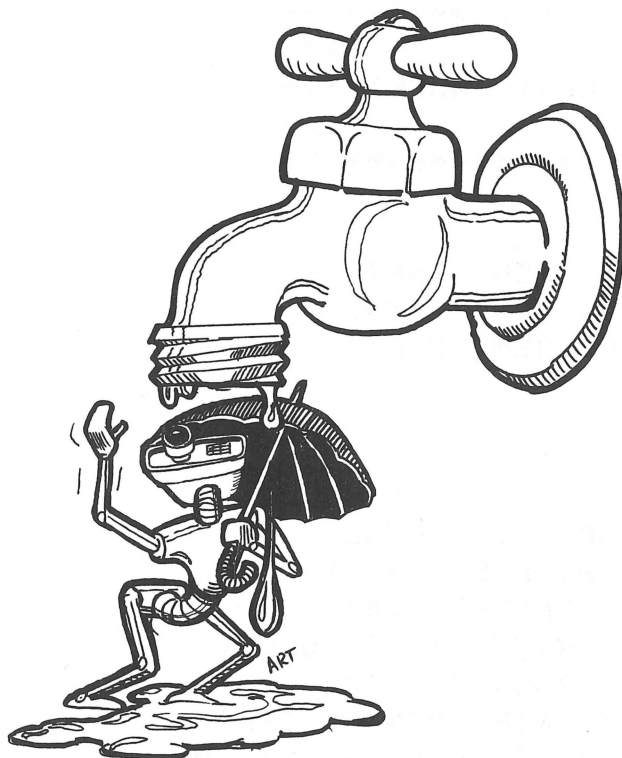
```

3125 POKE 0,50: POKE 1,10: CALL
      768
3130 IF X2 < > 25 THEN BD = -
      BD:BS = (Y2 - 00) - (Y2 - 00
      < 2) * 3: GOTO 3080
3140 BD = - BD:BS = (Y2 - 01) -
      (Y2 - 01 < 3) * 3: GOTO 3080

3150 COLOR= 0: PLOT BX,BY: COLOR=
      15: PLOT X2,Y2:BX = X2:BY =
      Y2: GOTO 3080
3500 RETURN
4000 :
4001 REM *** END
4002 :
4010 HOME : PRINT "THE GAME IS O
      VER !!!"
4020 PRINT "THE WINNER IS PLAYER
      # ";2 - (L2 = 0)
4030 RETURN

```





# Leaky Faucet

This is not a game. Rather, it is an excellent demonstration of how to use graphics. The idea is to simulate the effect of a dripping faucet. List through line 3035. Remember that color 0 is black. With that in mind, can you guess the function of line 3010? Line 3010 gives the color a random value between 0 and 14. The assorted colors are for the original droplet configuration. If this is not clear, change line 3016 to read: 3016 COLOR= 2. Enter and then RUN. Line 3030 draws the faucet. Experiment with these commands to confirm their function. Likewise, experiment with lines 3660-3717, and see if you can deduce their function.

```

10 REM *****
11 REM *** ***
12 REM *** LEAKY FAUCET ***
13 REM *** ***
14 REM *****
15 REM
16 REM
20 GOSUB 1000: REM INSTS
30 GOSUB 2000: REM SETUP
40 GOSUB 3000: REM PLAY!
60 END

1000 :
1001 REM *** INSTS
1002 :
1010 TEXT : HOME : NORMAL
1020 VTAB 3: HTAB 8: PRINT "***
    THE LEAKY FAUCET ***"
1030 VTAB 7: PRINT "THIS IS NOT
    REALLY A GAME, BUT AN
    EXTREMELY ENTERTAINING GRAPH
    ICS DEMO.": PRINT : PRINT "W
    E HOPE YOU ENJOY IT !!!"
1040 VTAB 23: INPUT "HIT RETURN
    WHEN READY TO CONTINUE : ";A
    NS$
1990 RETURN
2000 :
2001 REM *** SETUP
2002 :
2010 GR :NU = 200: HOME : PRINT
    : PRINT "<<< TURN PADDLE 0 T
    O MOVE FAUCET >>>": PRINT "<
    << HOLD BUTTON 0 TO START AG
    AIN >>>"
2990 RETURN
3000 :
3001 REM *** PLAY
3002 :
3010 COLOR= 0: FOR I = 0 TO 19: HLIN
    0,39 AT I: HLIN 0,39 AT 39 -
    I: NEXT : FOR I = 1 TO NU: COLOR=
    INT ( RND (1) * 15): PLOT (
    INT ( RND (1) * 38)) + 1, INT
    ( RND (1) * 20) + 20: NEXT I

```



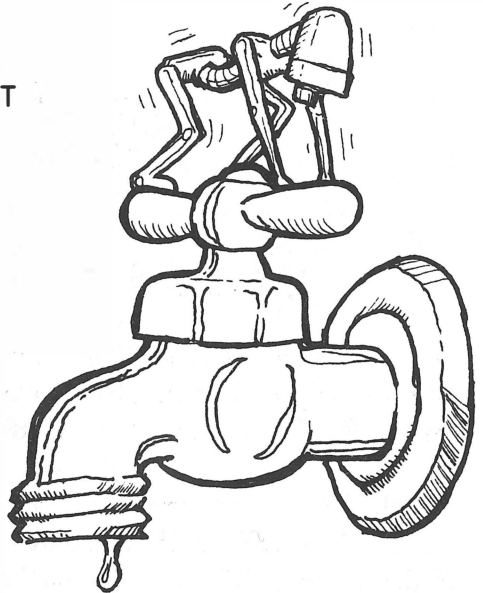
```

3030  COLOR= 4: HLIN 0,2 AT 5: HLIN
      0,2 AT 6: HLIN 0,2 AT 7: COLOR=
      1: HLIN 3,5 AT 4: HLIN 3,5 AT
      5: HLIN 3,5 AT 6: HLIN 3,5 AT
      7: COLOR= 6: PLOT 5,8: PLOT
      5,3: HLIN 3,7 AT 2
3110  I = 5
3120  SX = I:SY = 9:XX =  PEEK ( -
      16336) +  PEEK ( - 16336) +
      PEEK ( - 16336) +  PEEK ( -
      16336)
3130  COLOR= 15: PLOT SX,SY: GOSUB
      3600
3135  S2 = SY + 1: IF S2 > 39 THEN
      3200
3140  IF  SCRN( SX,S2) <  > 0 THEN
      3200
3145  COLOR= 0: PLOT SX,SY:SY = S
      2: GOTO 3130
3200  LR =  INT ( RND (1) * 2) * 2
      - 1: IF  SCRN( SX + LR,SY) <
      > 0 AND  SCRN( SX - LR,SY) <
      > 0 THEN 3500
3215  IF  SCRN( SX + LR,SY) <  >
      0 THEN LR =  - LR: GOTO 3215

3220  S2 = SX + LR: IF S2 < 1 OR S
      2 > 38 THEN 3500
3225  IF  SCRN( S2,SY) <  > 0 THEN
      3500
3230  COLOR= 0: PLOT SX,SY: SX = S
      2: COLOR= 15: PLOT SX,SY: GOSUB
      3600
3235  IF SY < 39 THEN  IF  SCRN(
      SX,SY + 1) = 0 THEN 3135
3240  GOTO 3220
3500  IF  PEEK ( - 16287) < 128 THEN
      3120
3510  GOTO 3010
3600  P =  PDL (0)
3605  IF P < 90 THEN 3700
3607  IF P < 180 THEN  RETURN
3640  IF I = 35 THEN  RETURN

```

```
3650  COLOR= 0: PLOT I - 2,2: PLOT  
      I - 2,4: PLOT I,8: PLOT I,3  
3655  I = I + 1  
3660  COLOR= 4: VLIN 5,7 AT I - 3  
  
3670  COLOR= 1: VLIN 4,7 AT I  
3680  COLOR= 6: PLOT I + 2,2: PLOT  
      I,3: PLOT I,8  
3690  RETURN  
3700  IF I = 5 THEN RETURN  
3710  COLOR= 0: PLOT I + 2,2: VLIN  
      3,8 AT I  
3715  I = I - 1  
3716  COLOR= 1: VLIN 4,7 AT I - 2  
  
3717  COLOR= 6: PLOT I - 2,2: PLOT  
      I,3: PLOT I,8  
3799  RETURN
```





# Match the Key

If you have ever played or ever seen Simon, then you will recognize this game. You must give the numeric equivalent of a lighted box sequence. There are a number of mnemonic devices which make it easier to recall a long string, but try to develop your own. A very good player can repeat a sequence of twenty boxes, and an expert can repeat a chain of thirty. The graphics in this program are interesting. Let's take a look. Lines 2020-2025 are responsible for the four brown squares being drawn. To verify this, change line 2021 so that it reads: 2021 COLOR = 1. When you run the program the four boxes will be red. To change the size and shape of the four boxes, experiment with line 2024. Experiment with any lines that have an unclear function.

```

10 REM *****
11 REM *** ***
12 REM *** MATCH THE KEY **
13 REM *** ***
14 REM *****
15 REM
16 REM
20 GOSUB 1000: REM INSTS
30 GOSUB 2000: REM SETUP
40 GOSUB 3000: REM PLAY!
50 GOSUB 4000: REM !END!
60 END
1000
1001 REM *** INSTS
1002
1010 TEXT : NORMAL : HOME
1020 VTAB 3: HTAB 9: PRINT "***
MATCH THE KEY ***"
1030 VTAB 7: PRINT "THIS GAME WI
LL TEST YOUR MEMORY. YOU
WILL BE SHOWN A SCREEN WITH
FOUR COLOREDBLOCKS ON IT. T
HE COMPUTER WILL LIGHT UP O
NE OF THE BLOCKS, AND SOUND
ITS CORRESPONDING TONE."

1035 PRINT
1040 PRINT "INPUT THE NUMBER OF
THE LIGHTED KEY/S. IF YOU A
RE CORRECT, THE COMPUTER WIL
L REPEAT THE SEQUENCE AND
ADD ANOTHER COLOR TO IT.
"

1045 PRINT
1050 PRINT "AFTER EACH ADDITION,
YOU MUST RETYPE THE ENTIRE S
EQUENCE. YOU ARE ALLOWED TH
REE MISTAKES PER GAME. "
1060 VTAB 22: INPUT "HIT RETURN
WHEN READY TO CONTINUE : ";A
NS$
1990 RETURN
2000

```

```

2001  REM  *** SETUP
2002  :
2010  DIM SEQ(50): FOR I = 1 TO 5
      0:SE(I) = INT ( RND (1) * 4
      ) + 1: NEXT I
2020  GR : HOME
2021  COLOR= 8
2022  FOR I = 5 TO 29 STEP 8
2023  FOR J = 29 TO 35
2024  HLIN I,I + 5 AT J
2025  NEXT J,I
2030  VTAB 21: FOR I = 5 TO 29 STEP
      8: HTAB I + 4: PRINT (I + 3)
      / 8;: NEXT
2040  POKE 768,173: POKE 769,48: POKE
      770,192: POKE 771,136: POKE
      772,208: POKE 773,4: POKE 77
      4,198: POKE 775,1: POKE 776,
      240
2045  POKE 777,8: POKE 778,202: POKE
      779,208: POKE 780,246: POKE
      781,166: POKE 782,0: POKE 78
      3,76: POKE 784,0: POKE 785,3
      : POKE 786,96
2990  RETURN
3000  :
3001  REM  *** PLAY
3002  :
3010  FOR I = 1 TO 50:SC = (I - 1
      ) - M: VTAB 22: HTAB 10: PRINT
      "MISSES: "M"      SCORE: "SC
3020  FOR J = 1 TO I: COLOR= 15: FOR
      K = 29 TO 35: HLIN SE(J) * 8
      - 3,SE(J) * 8 + 2 AT K: NEXT
      K: POKE 0, (5 - SE(J)) * 60: POKE
      1,50: CALL 768
3020  FOR J = 1 TO I: COLOR= 15: FOR
      K = 29 TO 35: HLIN SE(J) * 8
      - 3,SE(J) * 8 + 2 AT K: NEXT
      K: POKE 0,(5 - SE(J)) * 60: POKE
      1,50: CALL 768
3030  COLOR= 8: FOR K = 29 TO 35:
      HLIN SE(J) * 8 - 3,SE(J) *
      8 + 2 AT K: NEXT K: NEXT J

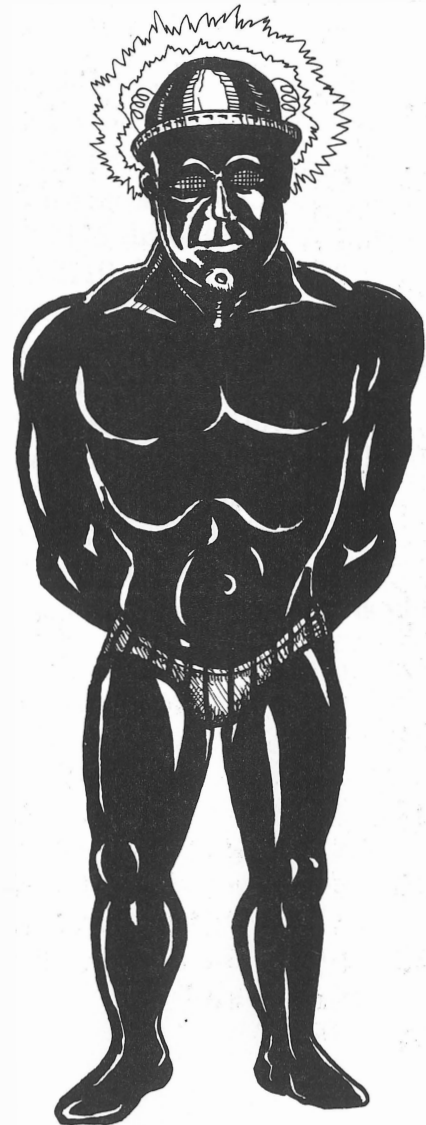
```

```

3100  FOR J = 1 TO I
3110  IF PEEK ( - 16384) < 128 THEN
3110
3120  X = PEEK ( - 16384): POKE
16368,0
3130  IF X < 177 OR X > 180 THEN
3110
3135  COLOR= 15: FOR K = 29 TO 35
: HLIN (X - 176) * 8 - 3,(X -
176) * 8 + 2 AT K: NEXT K: POKE
0,(5 - (X - 176)) * 60: POKE
1,50: CALL 768
3140  COLOR= 8: FOR K = 29 TO 35:
HLIN (X - 176) * 8 - 3,(X -
176) * 8 + 2 AT K: NEXT K
3150  IF X - 176 = SE(J) THEN 320
0
3155  POKE 0,250: POKE 1,100: CALL
768:M = M + 1: IF M < 3 THEN
3300
3160  RETURN
3200  NEXT J
3300  FOR PA = 1 TO 500: NEXT PA:
NEXT I: RETURN
3990  RETURN
4000
4001  REM *** END
4002
4010  HOME
4020  FOR J = 0 TO 39
4030  COLOR= INT ( RND (1) * 12)
* 1: HLIN 0,39 AT J
4031  COLOR= INT ( RND (1) * 12)
* 1: VLIN 0,39 AT 39 - J
4032  COLOR= INT ( RND (1) * 12)
* 1: HLIN 0,39 AT 39 - J
4033  COLOR= INT ( RND (1) * 12)
* 1: VLIN 0,39 AT J
4040  POKE 0,J * 2: POKE 1,5: CALL
768: NEXT J
4045  PRINT

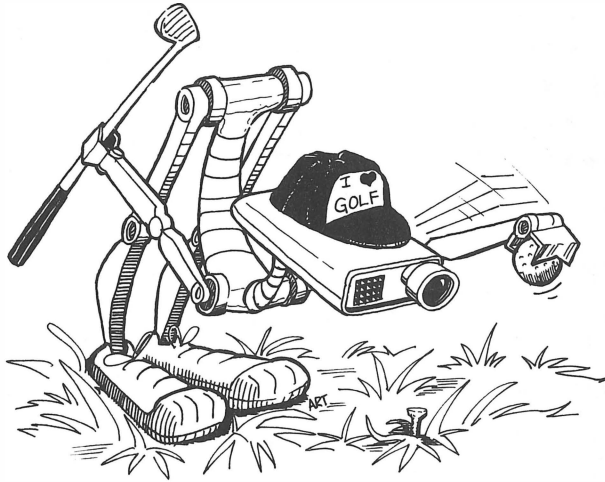
```

```
4050 IF M = 3 THEN PRINT "<<< I  
      'M SORRY YOU LOST . . .>>>": IF  
      SC < 1 THEN PRINT "YOU GOT  
      THEM ALL WRONG!!!": RETURN  
4055 PRINT "<<< YOU GOT ALL 50 0  
      F THEM !!! >>>": RETURN
```









# Miniature GOLF

This game is a simulation of miniature golf. There are hazards, obstacles, and unplayable lies, just as in the real thing. The graphics are interesting, and merit a closer look. Type: LIST -2060. Lines 2050-2058 draw the yellow-green background. For practice, change line 2050 so that it reads: 2050 COLOR = 6. Enter and run. Next, list through 2115. Lines 2105-2115 draw hole #1. Lines 2106-7-8 draw the red frame around the hole. Line 2110 draws the brown square which represents the hole. Line 2115 draws the white square which represents the ball. Line 2200 begins the graphics for hole #2. Line 2300 begins hole #3. Line 2400 begins hole #4, and so on through hole #9 (begins at 2900). It would be to your benefit to experiment with any of the lines (2000-2915) that are unclear. As with most programs, the hard part is moving the ball and charting its path. The ball is white, (color = 15), so look for statements preceded by 'COLOR = 15'. Look at statements 3110 thru 3124. The SCRN function returns the number of the color of X2,Y2 and therefore tells the program what color surface the ball has landed on. The SCRN function is very useful for detecting 'hits' in all sorts of action games.

```

10 REM *****
11 REM *** ***
12 REM ** MINIATURE GOLF **
13 REM *** ***
14 REM *****
15 REM
16 REM
20 GOSUB 1000: REM INSTS
30 GOSUB 2000: REM SETUP
40 GOSUB 3000: REM PLAY!
50 GOSUB 4000: REM !END!
60 END

1000 :
1001 REM *** INSTS
1002 :
1010 TEXT : NORMAL : HOME
1020 VTAB 1: HTAB 9: PRINT "***
      MINIATURE GOLF ***"
1030 VTAB 5: PRINT "WELCOME TO T
      HE CAPELLA COUNTRY CLUB.
      THIS BEAUTIFUL NINE-HOLE MIN
      IATURE GOLF COURSE IS OPEN A
      ND WAITING FOR YOU ! "
1035 PRINT
1040 PRINT "YOU SHOULD KNOW THE
      IDIOSYNCRASIES OF THE COUR
      SE BEFORE YOU BEGIN PLAY. "
1045 PRINT : PRINT "TO PUTT THE
      BALL, YOU HAVE TO INPUT
      WHICH DIRECTION YOU WANT TO
      AIM IT. THERE ARE EIGHT
      DIRECTIONS, SHOWN BELOW.YOUR
      BALL IS ASSUMED TO BE AT *
      , "
1050 PRINT : PRINT "
      2 1 8": PRINT " 3
      * 7": PRINT " 4
      5 6"
1060 VTAB 23: INPUT "HIT RETURN
      WHEN READY TO CONTINUE : ";A
      NS$

```

```

1070 HOME : VTAB 1: HTAB 9: PRINT
    "*** MINIATURE GOLF ***": VTAB
    5
1080 PRINT "THEN YOU MUST INPUT
    HOW HARD TO HIT THE BALL. T
    HE SPEED SHOULD BE SOME NUMB
    ER BETWEEN 0.00 AND 5.00.
    FOR EXAMPLE, YOU COULD HIT TH
    E BALL A RELATIVE SPEED OF
    3.2, "
1085 PRINT
1090 PRINT "IT WILL TAKE A FEW T
    RIES BEFORE YOU GET THE FEEL
    OF HOW HARD TO PUTT THE BAL
    L, "
1095 PRINT
1100 PRINT "ALSO, THERE ARE FOUR
    TYPES OF HAZARDS ON THE COUR
    SE. YOU SHOULD BE AWARE OF
    WHAT THEY ARE AND WHAT AFFECT
    THEY HAVE ON YOU AND YOUR
    BALL, "
1110 VTAB 23: INPUT "HIT RETURN
    WHEN READY TO CONTINUE : ";A
    NS$
1120 HOME : VTAB 1: HTAB 9: PRINT
    "*** MINIATURE GOLF ***": VTAB
    5
1130 PRINT "BLOCKS: THESE ARE LI
    KE WALLS. YOU MUST PUTT ARO
    UND THEM, "
1135 PRINT
1140 PRINT "YELLOW TRAPS REPRESE
    NT SAND. YOUR BALL CANNOT P
    ENETRATE THROUGH A SAND TRAP
    THE PENALTY FOR LANDING
    IN SAND IS ONE STROKE, "
1145 PRINT
1150 PRINT "WATER: LIKE SAND, TH
    ESE BLUE HAZARDS WILL SLO
    W AND STOP YOUR BALL. THE
    PENALTY FOR LANDING IN W
    ATER IS ONE STROKE, "

```

```

1155 PRINT
1160 PRINT "UNEVEN SURFACES: THE
      SE ORANGE HAZARDS CAUSE TH
      E BALL TO ROLL IN A DIRECTIO
      N WHICH IS UNPREDICTABLE.
      THERE IS NO PENALTY FOR
      HITTING THIS HAZARD. "
1170 VTAB 23: INPUT "PRESS RETUR
      N WHEN READY TO CONTINUE : "
      ;ANS$
1180 HOME : VTAB 1: HTAB 9: PRINT
      "*** MINIATURE GOLF ***": VTAB
      5
1190 PRINT "YOU ARE TRYING TO SI
      NK THE BALL IN AS FEW TRIE
      S AS POSSIBLE. THE HOLE IS
      THE BROWN SQUARE. "
1195 PRINT
1200 PRINT "IF YOU HIT THE BALL
      TOO HARD, IT WILL JUMP OVE
      R THE HOLE AND CONTINUE ROLL
      ING. IT MAY ALSO CHANGE DIREC
      TION, SO BE SURE TO HIT THE
      BALL JUST HARD ENOUGH. "
1210 VTAB 23: INPUT "HIT RETURN
      WHEN READY TO CONTINUE : ";A
      NS$
1990 RETURN
2000 :
2001 REM *** SETUP
2002 :
2005 BELL$ = CHR$ (7)
2010 DIM HA(9,5): FOR I = 1 TO 9
      : FOR J = 1 TO 5: READ HA(I,
      J): NEXT J,I
2011 DATA 0,0,0,0,2
2012 DATA 1,0,0,0,3
2013 DATA 0,1,0,0,3
2014 DATA 1,1,0,0,3
2015 DATA 0,0,0,1,3
2016 DATA 0,0,1,1,3
2017 DATA 1,0,1,0,3
2018 DATA 0,1,1,0,4
2019 DATA 0,0,1,1,3

```



```

2020  DIM DI(8,2): FOR I = 1 TO 8
      READ DI(I,1),DI(I,2): NEXT
      : DATA    0,-1,  -1,-1,  -1,0
        -1,1,  0,1,  1,1,  1,0,
        1,-1
2045  GR : HOME : RETURN
2050  COLOR= 12
2052  FOR I = 0 TO 19
2054  HLIN 0,39 AT I: HLIN 0,39 AT
      39 - I
2056  VLIN 0,39 AT I: VLIN 0,39 AT
      39 - I
2058  NEXT : RETURN
2100  REM *** HOLE 1
2105  GOSUB 2050
2106  COLOR= 1
2107  HLIN 10,24 AT 6: HLIN 10,24
      AT 33
2108  VLIN 6,33 AT 10: VLIN 6,33 AT
      24
2110  COLOR= 8: PLOT 17,9
2115  BY = 32:BX = INT ( RND (1) *
      11) + 12: COLOR= 15: PLOT BX
      ,BY: RETURN
2200  REM *** HOLE 2
2205  GOSUB 2050: COLOR= 1: HLIN
      10,24 AT 33: VLIN 6,33 AT 10
      : VLIN 16,33 AT 24: HLIN 10,
      35 AT 6: HLIN 24,35 AT 16: VLIN
      6,16 AT 35
2207  COLOR= 2: VLIN 18,19 AT 20:
      VLIN 17,21 AT 21: VLIN 17,2
      3 AT 22: VLIN 16,26 AT 23
2210  COLOR= 8: PLOT 32,11
2215  BY = 32:BX = INT ( RND (1) *
      11) + 12: COLOR= 15: PLOT BX
      ,BY: RETURN
2300  REM *** HOLE 3
2305  GOSUB 2050: COLOR= 1: HLIN
      24,35 AT 6: HLIN 10,24 AT 15
      : HLIN 24,35 AT 24: HLIN 10,
      24 AT 33: VLIN 6,15 AT 24: VLIN
      15,33 AT 10: VLIN 24,33 AT 2
      4: VLIN 6,24 AT 35

```

```

2307  COLOR= 13: PLOT 30,23: VLIN
      22,23 AT 31: VLIN 20,23 AT 3
      2: VLIN 19,23 AT 33: VLIN 17
      ,23 AT 34
2310  COLOR= 8: PLOT 32,9
2315  BY = 32:BX = INT ( RND (1) *
      11) + 12: COLOR= 15: PLOT BX
      ,BY: RETURN
2400  REM *** HOLE 4
2405  GOSUB 2050: COLOR= 1: HLIN
      5,35 AT 6: HLIN 15,25 AT 15:
      HLIN 25,35 AT 24: HLIN 5,15
      AT 33: VLIN 6,33 AT 5: VLIN
      15,33 AT 15: VLIN 15,24 AT 2
      5: VLIN 6,24 AT 35
2407  COLOR= 13: PLOT 15,7: VLIN
      7,8 AT 16: VLIN 7,9 AT 17: VLIN
      7,9 AT 18: VLIN 7,8 AT 19: PLOT
      20,7
2408  COLOR= 2: PLOT 15,14: VLIN
      13,14 AT 16: VLIN 13,14 AT 1
      7: VLIN 13,14 AT 18: VLIN 13
      ,14 AT 19: VLIN 13,14 AT 20:
      VLIN 12,14 AT 21: VLIN 11,1
      4 AT 22
2410  COLOR= 8: PLOT 32,21
2415  BY = 32:BX = INT ( RND (1) *
      9) + 6: COLOR= 15: PLOT BX,B
      Y: RETURN
2500  REM *** HOLE 5
2505  GOSUB 2050: COLOR= 1: HLIN
      10,24 AT 6: HLIN 10,24 AT 33
      : VLIN 6,33 AT 10: VLIN 6,33
      AT 24: HLIN 13,21 AT 17: VLIN
      12,17 AT 13: VLIN 12,17 AT 2
      1
2510  COLOR= 8: PLOT 17,9
2515  BY = 32:BX = INT ( RND (1) *
      11) + 12: COLOR= 15: PLOT BX
      ,BY: RETURN
2600  REM *** HOLE 6

```

```

2605  GOSUB 2050: COLOR= 1: HLIN
      10,24 AT 33: VLIN 6,33 AT 10
      : VLIN 16,33 AT 24: HLIN 10,
      35 AT 6: HLIN 24,35 AT 16: VLIN
      6,16 AT 35
2607  HLIN 26,29 AT 9: HLIN 26,29
      AT 13: VLIN 9,13 AT 26
2608  COLOR= 9: VLIN 13,18 AT 21:
      VLIN 13,18 AT 22: VLIN 13,1
      8 AT 23: VLIN 13,15 AT 24: VLIN
      14,15 AT 25: PLOT 26,15
2610  COLOR= 8: PLOT 32,11
2615  BY = 32:BX = INT ( RND (1) *
      11) + 12: COLOR= 15: PLOT BX
      ,BY: RETURN
2700  REM *** HOLE 7
2705  GOSUB 2050: COLOR= 1: HLIN
      5,15 AT 6: HLIN 15,35 AT 12:
      HLIN 5,25 AT 24: HLIN 25,35
      AT 33: VLIN 6,24 AT 5: VLIN
      6,12 AT 15: VLIN 24,33 AT 25
      : VLIN 12,33 AT 35
2707  COLOR= 2: VLIN 16,23 AT 6: VLIN
      17,23 AT 7: VLIN 18,23 AT 8:
      VLIN 18,23 AT 9: VLIN 20,23
      AT 10: PLOT 11,23
2708  COLOR= 9: VLIN 13,28 AT 31:
      VLIN 13,28 AT 32: VLIN 13,2
      8 AT 33: VLIN 13,28 AT 34: HLIN
      25,30 AT 13: HLIN 25,30 AT 1
      4: HLIN 25,30 AT 15
2710  COLOR= 8: PLOT 10,9
2715  BY = 32:BX = INT ( RND (1) *
      7) + 26: COLOR= 15: PLOT BX,
      BY: RETURN
2800  REM *** HOLE 8
2805  GOSUB 2050: COLOR= 1: HLIN
      5,35 AT 6: HLIN 15,25 AT 15:
      HLIN 5,15 AT 24: HLIN 25,35
      AT 33: VLIN 6,24 AT 5: VLIN
      15,24 AT 15: VLIN 15,33 AT 2
      5: VLIN 6,33 AT 35

```

```

2807  COLOR= 9: VLIN 7,8 AT 25: VLIN
      7,9 AT 26: VLIN 7,11 AT 27: VLIN
      7,11 AT 28: VLIN 7,12 AT 29:
      VLIN 7,12 AT 30: VLIN 7,17 AT
      31: VLIN 7,18 AT 32: VLIN 7,
      19 AT 33: VLIN 7,20 AT 34
2808  COLOR= 13: VLIN 11,13 AT 8:
      VLIN 10,15 AT 9: VLIN 9,15 AT
      10: VLIN 9,16 AT 11: VLIN 9,
      16 AT 12
2810  COLOR= 8: PLOT 10,21
2815  BY = 32:BX = INT ( RND (1) *
      7) + 26: COLOR= 15: PLOT BX,
      BY: RETURN
2900  REM *** HOLE 9
2905  GOSUB 2050: COLOR= 1: HLIN
      5,35 AT 6: HLIN 5,25 AT 18: HLIN
      25,35 AT 33: VLIN 6,18 AT 5:
      VLIN 18,33 AT 25: VLIN 6,33
      AT 35
2907  HLIN 8,11 AT 9: VLIN 9,15 AT
      11
2908  COLOR= 9: HLIN 25,34 AT 7: HLIN
      27,34 AT 8: HLIN 29,34 AT 9:
      HLIN 30,34 AT 10: HLIN 31,3
      4 AT 11
2910  COLOR= 8: PLOT 8,12
2915  BY = 32:BX = INT ( RND (1) *
      7) + 26: COLOR= 15: PLOT BX,
      BY: RETURN
3000  :
3001  REM *** PLAY
3002  :
3010  FOR HO = 1 TO 9:BC = 12
3020  ON HO GOSUB 2100,2200,2300,
      2400,2500,2600,2700,2800,290
      0
3030  HOME : PRINT "HOLE NUMBER:
      "HO" PAR: "HA(HO,5)" SCORE:
      "SC
3031  PRINT "TRAPS: "; IF HA(HO,
      1) THEN PRINT "WATER ";
3032  IF HA(HO,2) THEN PRINT "SA
      ND ";

```

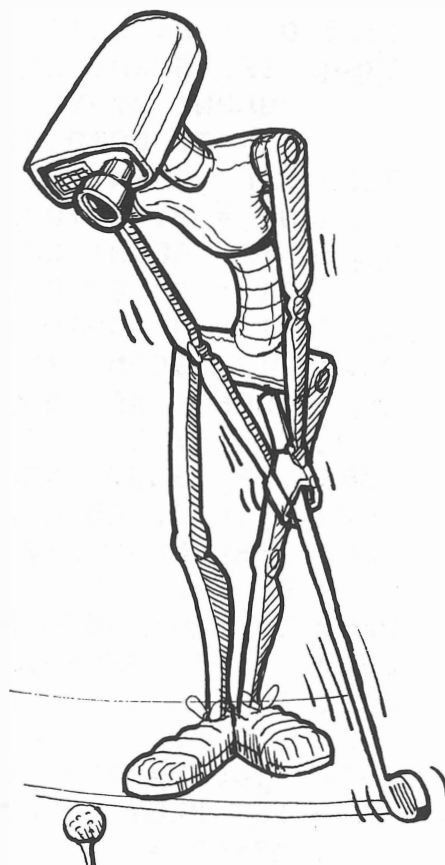


```

3033 IF HA(HO,3) THEN PRINT "UN
      EVEN ";
3034 IF HA(HO,4) THEN PRINT "BL
      OCKS ";
3035 VTAB 23: HTAB 1: CALL - 95
      8: INPUT "DIRECTION (1-8): "
      ;DIR
3040 IF DIR < 1 OR DIR > 8 OR DI
      R < > INT (DIR) THEN VTAB
      23: CALL - 958: PRINT "THE
      DIRECTION IS FROM 1 TO 8 ...
      ": FOR PA = 1 TO 1500: NEXT
      PA: GOTO 3035
3045 VTAB 23: CALL - 958: INPUT
      "SPEED (0-5): ";SP
3050 IF SP < 0 OR SP > 5 THEN VTAB
      23: CALL - 958: PRINT "THE
      SPEED IS FROM 0 TO 5 ...": FOR
      PA = 1 TO 1500: NEXT PA: GOTO
      3045
3055 UF = 0: REM CLEAR UNEVEN FL
      AG
3056 TF = 0: REM CLEAR TRAP FLAG

3100 X2 = BX + DI(DI,1):Y2 = BY +
      DI(DI,2)
3110 IF SCRN( X2,Y2) = 12 THEN
      COLOR= BC: PLOT BX,BY: COLOR=
      15: PLOT X2,Y2:BX = X2:BY =
      Y2:BC = 12: GOTO 3900
3120 IF SCRN( X2,Y2) = 01 THEN
      3200
3121 IF SCRN( X2,Y2) = 02 THEN
      3300
3122 IF SCRN( X2,Y2) = 13 THEN
      3400
3123 IF SCRN( X2,Y2) = 09 THEN
      3500
3124 IF SCRN( X2,Y2) = 08 THEN
      3600

```



```

3200  IF DI = 1 OR DI = 3 THEN DI
      = DI + 4: GOTO 3100
3201  IF DI = 7 OR DI = 5 THEN DI
      = DI - 4: GOTO 3100
3210  ON DI / 2 GOTO 3220,3240,32
      60,3280
3220  IF  SCRN( X2 + 1,Y2) = 1 AND
      SCRN( X2,Y2 + 1) = 1 THEN D
      I = 6: GOTO 3100
3225  IF  SCRN( X2 + 1,Y2) = 1 THEN
      DI = 4: GOTO 3100
3230  IF  SCRN( X2,Y2 + 1) = 1 THEN
      DI = 8: GOTO 3100
3235  DI = 6: GOTO 3100
3240  IF  SCRN( X2 + 1,Y2) = 1 AND
      SCRN( X2,Y2 - 1) = 1 THEN D
      I = 8: GOTO 3100
3245  IF  SCRN( X2 + 1,Y2) = 1 THEN
      DI = 2: GOTO 3100
3250  IF  SCRN( X2,Y2 - 1) = 1 THEN
      DI = 6: GOTO 3100
3255  DI = 8: GOTO 3100
3260  IF  SCRN( X2 - 1,Y2) = 1 AND
      SCRN( X2,Y2 - 1) = 1 THEN D
      I = 2: GOTO 3100
3265  IF  SCRN( X2 - 1,Y2) = 1 THEN
      DI = 8: GOTO 3100
3270  IF  SCRN( X2,Y2 - 1) = 1 THEN
      DI = 4: GOTO 3100
3275  DI = 2: GOTO 3100
3280  IF  SCRN( X2 - 1,Y2) = 1 AND
      SCRN( X2,Y2 + 1) = 1 THEN D
      I = 4: GOTO 3100
3285  IF  SCRN( X2 - 1,Y2) = 1 THEN
      DI = 6: GOTO 3100
3290  IF  SCRN( X2,Y2 + 1) = 1 THEN
      DI = 2: GOTO 3100
3295  DI = 4: GOTO 3100
3300  IF TF THEN 3310
3305  TF = 3: SC = SC + 1: REM  TRA
      P FLAG
3310  COLOR= BC: PLOT BX,BY: COLOR=
      15: PLOT X2,Y2: BX = X2: BY =
      Y2: BC = 2

```

```

3320 GOTO 3900
3400 IF TF THEN 3410
3405 TF = 3: SC = SC + 1: REM TRA
    P FLAG
3410 COLOR= BC: PLOT BX,BY: COLOR=
    15: PLOT X2,Y2: BX = X2: BY =
    Y2: BC = 13
3420 GOTO 3900
3500 IF UF THEN 3520
3505 UF = 1: REM UNEVEN FLAG, HA
    VE WE ROLLED BALL OFF COURSE
    YET ?...
3510 DI = DI + INT ( RND (1) * 2
    ) * 2 - 1
3515 IF DI = 0 THEN DI = 8
3516 IF DI = 9 THEN DI = 1
3520 COLOR= BC: PLOT BX,BY: COLOR=
    15: PLOT X2,Y2: BX = X2: BY =
    Y2: BC = 9: GOTO 3900
3600 COLOR= 12: PLOT BX,BY: COLOR=
    15: PLOT X2,Y2: COLOR= 8: PLOT

3602 DI = DI + INT ( RND (1) * 2
    ) * 2 - 1
3604 IF DI = 0 THEN DI = 8
3605 SP = SP - .4: IF SP > 0 THEN
    X2 = X2 + DI(DI,1): Y2 = Y2 +
    DI(DI,2): GOTO 3110
3606 IF DI = 9 THEN DI = 1
3610 PRINT BELL$BELL$BELL$: SC =
    SC + 1: GOTO 3990
3900 XX = PEEK ( - 16336) - PEEK
    ( - 16336)
3902 IF TF > 0 THEN TF = TF - 1:
    IF TF = 0 THEN 3910
3905 SP = SP - .2: IF SP > 0 THEN
    3100
3910 SC = SC + 1: GOTO 3030
3990 NEXT HO: RETURN
4000
4001 REM *** END
4002

```

```
4010  TEXT : HOME : VTAB 3: HTAB
      9: PRINT "*** MINIATURE GOLF
      ***": VTAB 7
4020  PRINT BELL$BELL$BELL$"THE G
      AME IS OVER !!!"
4022  PRINT
4025  PRINT "ON THE PAR 27 COURSE
      , YOU SHOT ": PRINT "A ROUND
      OF "SC". THAT IS AN": PRINT
      "AVERAGE OF "SC / 9" SHOTS P
      ER HOLE. "
4030  VTAB 22: PRINT "HOPE YOU EN
      JOYED THE GAME! "
4990  RETURN
```

# Moving Target

This is a one-man paddle game. The object is to shoot the moving targets. Different colored targets are worth different point scores. For the most part, this game is a measure of timing, but there is also a little luck involved. If you read through MUBBLE CHASE and understood it all, then these short graphic hints and explanations may seem mundane. First, load the program. Type: LIST-2030. In terms of drawing the original game setup, lines 2020 through 2024 do the majority of the work. 2020 instructs the computer to switch from the text mode into the GGraphs (color) mode. 2021 starts a loop consisting of ten individual loops (passes). 2022 will change the color each time I changes. 2023 draws the lines that will comprise the launching pad (a collection of ten lines). To get a better idea of line 2021, type: 2021 FOR I = 1 TO 11 STEP 2. Check line 3547. You will note that the first of the two instructions sets the color to 15 (white). Since the missile that you launch is the only all-white figure you see, it follows that SX,SY are the missile's X,Y coordinates.

```

10 REM *****
11 REM *** ***
12 REM *** MOVING TARGET ***
13 REM *** ***
14 REM *****
15 REM
16 REM
20 GOSUB 1000: REM INSTS
30 GOSUB 2000: REM SETUP
40 GOSUB 3000: REM PLAY!
50 GOSUB 4000: REM !END!
60 END

1000 :
1001 REM *** INSTS
1002 :
1010 TEXT : NORMAL : HOME
1020 VTAB 3: HTAB 10: PRINT "***
MOVING TARGET ***"
1030 VTAB 7: PRINT "IN THIS GAME
, YOU CONTROL A MISSLE'S
LAUNCH SITE. THE LAUNCHER I
S CONTROLLED BY PADDLE 0.,
"
1040 PRINT : PRINT "PRESSING THE
BUTTON ON THE PADDLE WILL
RELEASE A ROCKET. TRY TO HI
T ONE OF THE THREE MOVING TAR
GETS ABOVE YOU."
1050 PRINT : PRINT "DIFFERENT CO
LORS ARE WORTH DIFFERENT
AMOUNTS OF POINTS. SHOOT FOR
THE BEST SCORE."
1060 VTAB 23: INPUT "HIT RETURN
WHEN READY TO CONTINUE : ";A
NS$
1990 RETURN
2000 :
2001 REM *** SETUP
2002 :
2010 DIM TA(3,4)
2015 FOR I = 1 TO 3:TA(I,1) = INT
( RND (1) * 37):TA(I,2) = INT
( RND (1) * (4 + (I * 2))) +
1:TA(I,0) = INT ( RND (1) *
2) * 2 - 1

```

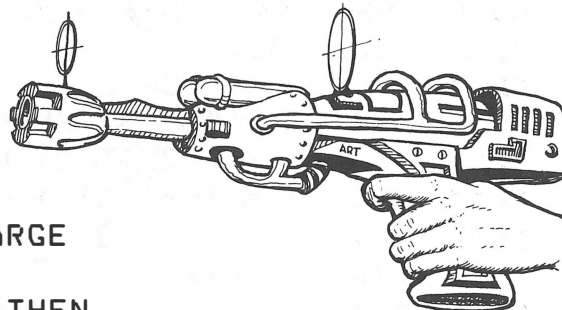
```

2016 TA(I,3) = 0:TA(I,4) = INT (
      RND (1) * 20) + 1: NEXT
2020 GR
2021 FOR I = 1 TO 10
2022 COLOR= I
2023 HLIN (I - 1) * 4,(I - 1) *
      4 + 3 AT 39
2024 NEXT
2030 HOME : PRINT " 10  20  30
      40  50  60  70  80  90 100"

2990 RETURN
3000 :
3001 REM *** PLAY
3002 :
3010 REM
3020 GOSUB 3600: REM MOVE TARGE
      TS
3040 NM = NM + 1: IF NM < 200 THEN
      3020
3500 P = INT ( PDL (0) / 6.5): IF
      PP = P THEN 3520
3510 COLOR= 0: VLIN 37,38 AT PP:
      COLOR= 12: VLIN 37,38 AT P:
      PP = P
3520 IF F THEN 3540
3525 IF PEEK ( - 16287) < 128 THEN
      RETURN
3530 F = 1: SX = PP: SY = 36
3540 COLOR= 0: PLOT SX,SY: SY = S
      Y - 1
3542 IF PEEK ( - 16287) > 127 THEN
      3530
3545 IF SY < 0 THEN F = 0: RETURN
3546 IF SCRN( SX,SY) < > 0 THEN
      3551
3547 COLOR= 15: PLOT SX,SY: RETURN

3551 Z = (SY > 22) + (SY > 12) +
      (SY > 2)
3552 PT = PT + SCRN( SX,SY) * (4
      - Z)
3553 F = 0:TA(Z,3) = 0:TA(Z,4) =
      INT ( RND (1) * 20) + 1

```




```

3554 TA(Z,0) = INT ( RND (1) * 2
      ) * 2 - 1:TA(Z,2) = INT RND
      (1) * (4 + (I * 2))) + 1
3555 XX = PEEK ( - 16336) + PEEK
      ( - 16336) - PEEK ( - 16336
      ) + PEEK ( - 16336)
3560 HOME : PRINT : PRINT "<<< Y
      OUR SCORE IS "PT" >>>"
3565 RETURN
3600 FOR I = 1 TO 3
3602 GOSUB 3500
3605 COLOR= 0: FOR J = I * 10 -
      5 TO I * 10 - 3: HLINE TA(I,1
      ),TA(I,1) + 2 AT J: NEXT
3615 TA(I,1) = TA(I,1) + TA(I,0)
3616 TA(I,3) = TA(I,3) + 1: IF TA
      (I,3) = TA(I,4) THEN TA(I,3)
      = 0:TA(I,4) = INT ( RND (1
      ) * 20) + 1:TA(I,0) = INT (
      RND (1) * 2) * 2 - 1:TA(I,2
      ) = INT ( RND (1) * (4 + (I
      * 2))) + 1
3621 IF TA(I,1) < 0 THEN TA(I,1)
      = 36
3622 IF TA(I,1) > 36 THEN TA(I,1
      ) = 0
3625 COLOR= TA(I,2): FOR J = I *
      10 - 5 TO I * 10 - 3: HLINE T
      A(I,1),TA(I,1) + 2 AT J: NEXT
      NEXT I: RETURN
4000
4001 REM *** END
4002
4010 HOME : PRINT "THE GAME IS O
      VER !!!": PRINT "YOUR FINAL
      SCORE IS "PT
4990 RETURN

```





This one-man paddle game is a test of dexterity. The challenge is to maneuver the paddle so that the flying points do not collide with you. To be sure, there is some luck involved. There is an abundance of easy to understand GGraphics in this program, so it is time to begin looking at them. Line 2010 draws a white perimeter around the game board (field of play). To better understand the function of 3040 and 3041, type: 3040 COLOR=2. When you run the program, all of the points will be blue (COLOR=2). Line 3041 draws the points at their new X,Y coordinates (BP(K,1),BP(K,2)). Back at line 3020, the old X,Y coordinates of each point are blacked out (COLOR= 0). If you do not perceive the significance of this line, type: 3020 and return. Now when you run the program, the path of each point will be seen.

```

10 REM *****
11 REM *** ***
12 REM *** POINT ATTACK ***
13 REM *** ***
14 REM *****
15 REM
16 REM
20 GOSUB 1000: REM INSTS
30 GOSUB 2000: REM SETUP
40 GOSUB 3000: REM PLAY!
50 GOSUB 4000: REM !END!
60 END
1000 :
1001 REM *** INSTS
1002 :
1010 TEXT : NORMAL : HOME
1020 VTAB 3: HTAB 10: PRINT "***
    POINT ATTACK ***"
1030 VTAB 7: PRINT "IN THIS GAME
    YOU WILL CONTROL THE MOVE-
    MENT OF A PADDLE IN THE MIDD
    LE OF THE SCREEN BY MOVING
    PADDLE 0."
1040 PRINT : PRINT "THERE WILL B
    E A FLYING POINT THAT WILL
    BOUNCE AROUND ON THE WALLS.
    THE POINT WILL TRY TO HIT
    YOU, BUT DON'T LET IT."
1050 PRINT : PRINT "AFTER AWHILE
    , ANOTHER POINT WILL BE
    ADDED TO THE FIELD OF PLAY,
    AND YOUR PADDLE WILL GROW
    LARGER. AVOID BEING HIT
    FOR AS LONG AS YOU CAN."
1060 VTAB 23: INPUT "HIT RETURN
    WHEN READY TO CONTINUE : ";A
    NS$
1990 RETURN
2000 :
2001 REM *** SETUP
2002 :
2010 GR : COLOR= 15: HLIN 0,39 AT
    0: HLIN 0,39 AT 39: VLIN 0,3
    9 AT 0: VLIN 0,39 AT 39

```

```

2020 DIM BP(10,2),BD(10,2)
2030 FOR I = 1 TO 10:BP(I,1) = 2
    :BP(I,2) = INT ( RND (1) *
37) + 2:BD(I,1) = 1:BD(I,2) =
    INT ( RND (1) * 2) * 2 - 1:
    NEXT
2040 X = 20
2105 POKE 768,173: POKE 769,48: POKE
770,192: POKE 771,136: POKE
772,208: POKE 773,4: POKE 77
4,198: POKE 775,1: POKE 776,
240
2110 POKE 777,8: POKE 778,202: POKE
779,208: POKE 780,246: POKE
781,166: POKE 782,0: POKE 78
3,76: POKE 784,0: POKE 785,3
: POKE 786,96
2990 RETURN
3000 :
3001 REM *** PLAY
3002 :
3010 FOR H = 1 TO 30:X = X - 1: HOME
    : PRINT : PRINT "<<< SCORE =
==> "H" >>>": FOR J = 1 TO 2
5
3015 I = H: IF I > 10 THEN I = 10

3016 FOR K = 1 TO I
3020 COLOR= 0: PLOT BP(K,1),BP(K
,2)
3025 BP(K,1) = BP(K,1) + BD(K,1):
BP(K,2) = BP(K,2) + BD(K,2)
3030 IF BP(K,1) = 1 OR BP(K,1) =
38 THEN BD(K,1) = - BD(K,1)
: POKE 0, INT ( RND (1) * 25
6): POKE 1,10: CALL 768
3031 IF BP(K,2) = 1 OR BP(K,2) =
38 THEN BD(K,2) = - BD(K,2)
: POKE 0, INT ( RND (1) * 25
6): POKE 1,10: CALL 768
3035 IF SCRNB( BP(K,1),BP(K,2)) =
15 THEN RETURN
3040 COLOR= K

```

```

3041  PLOT BP(K,1),BP(K,2)
3045  IF X < 1 THEN X = 1
3050  COLOR= 0: VLIN X,X + H + 2 AT
      20
3052  P = PDL (0): IF P > 128 THEN
      X = X + 1: GOTO 3056
3055  X = X - 1
3056  IF X > 38 - (H + 2) THEN X =
      38 - (H + 2)
3057  IF X < 1 THEN X = 1
3060  COLOR= 15: VLIN X,X + H + 2
      AT 20
3070  POKE 0,2: POKE 1,1: CALL 76
      8
3099  NEXT K,J,H
3990  RETURN
4000 :
4001  REM *** END
4002 :
4010  HOME : PRINT "THE GAME IS O
      VER !!!"
4020  PRINT "YOU'RE SCORE IS "H",
      CONGRATULATIONS !!!"
4990  RETURN

```

In this game you try to avoid being captured by the killer robots. Actually, escaping from the robot's relentless pursuit is most difficult. Because the pursuit is entirely pre-determined, it might be a good idea to chart your course before you make your first move. Starting at line 2100, the text mode is completed and the graphics mode is begun. Remember, once you are in the graphics mode, the HOME command only clears the bottom four rows, which are reserved for text. Note that line 2090 sets all values of  $FI\%(I,0)$  equal to two. Line 2095 does the same thing to  $FI(0,I)$ . Now, when line 2110 is executed, a deep blue perimeter is drawn ( $COLOR = FI(I,J)$ ). Line 2075 sets  $FI(OX,OY)$  to 15. Line 3025 sets the color to 15 (white). Since you are represented by the white square, it would be understandable for you to experiment with these two instructions to see what and how they function.

```

10 REM *****
11 REM *** ***
12 REM *** ROBOT CHASE ***
13 REM *** ***
14 REM *****
15 REM
16 REM
20 GOSUB 1000: REM INSTS
30 GOSUB 2000: REM SETUP
40 GOSUB 3000: REM PLAY!
50 GOSUB 4000: REM !END!
60 END
1000 :
1001 REM *** INSTS
1002 :
1010 TEXT : NORMAL : HOME
1020 VTAB 2: HTAB 10: PRINT "***
      ROBOT CHASE ***"
1030 VTAB 5: PRINT "IN ROBOT CHA
      SE, YOU ARE AN EXPLORER WHO
      HAS LANDED HIS SPACESHIP ON
      A HOSTILE PLANET."
1040 PRINT : PRINT "SEVERAL PROT
      ECTOR ROBOTS ARE TRYING TO
      CAPTUE YOU. IF YOU CAN REAC
      H A BASE, YOU WILL BE SAFE
      BEHIND ITS PROTECTIVE FORC
      E FIELD."
1050 PRINT : PRINT "HERE'S HOW T
      HINGS WORK: ": PRINT : PRINT
      " BLUE - AN EXPLOSIVE FEN
      CE (BAD!)"
1060 PRINT " WHITE - YOU"
1070 PRINT " GREEN - ATTACKING
      ROBOT (BAD!)"
1080 PRINT " ORANGE- PROTECTIV
      E BASE (GOOD!!!)"
1090 VTAB 23: INPUT "HIT RETN
      WHEN READY TO CONTINUE : ";A
      NS$
1100 HOME : VTAB 11
1110 PRINT " 3 2 1"
1111 PRINT " I/ TH
      IS IS YOUR CHOICE"

```

```

1112 PRINT "      4-+-8      OF
      MOVEMENT"
1113 PRINT "      /I      TH
      ROUGH THE MAZE"
1114 PRINT "      5 6 7"
1120 VTAB 23: INPUT "HIT RETURN
      WHEN READY TO CONTINUE : ";A
      NS$
1990 RETURN
2000 :
2001 REM *** SETUP
2002 :
2010 DIM FI%(21,11)
2015 DEF FN R(X) = INT ( RND (
      1) * X) + 1
2020 NR = 4 + FN R(5): DIM RO%(9
      ,2)
2021 NO = INT ((NR - 4) / 2) + 1

2025 FOR I = 1 TO NR
2030 RX = FN R(20):RY = FN R(10
      )
2035 IF FI%(RX,RY) THEN 2030
2040 FI%(RX,RY) = 4:RO%(I,1) = RX
      :RO%(I,2) = RY: NEXT I
2045 FOR I = 1 TO NO
2050 OX = FN R(20):OY = FN R(10
      )
2055 IF FI%(OX,OY) THEN 2050
2060 FI%(OX,OY) = 9: NEXT I
2070 YX = FN R(20):YY = FN R(10
      )
2075 FI%(YX,YY) = 15
2090 FOR I = 0 TO 21:FI%(I,0) =
      2:FI%(I,11) = 2: NEXT I
2095 FOR I = 0 TO 11:FI%(0,I) =
      2:FI%(21,I) = 2: NEXT I
2100 GR : HOME
2110 FOR I = 0 TO 21: FOR J = 0 TO
      11: COLOR= FI%(I,J): PLOT I +
      9,J + 14: NEXT J,I

```

```

2200  DIM DI(8,2): FOR I = 1 TO 8
      : READ DI(I,1),DI(I,2): NEXT
      I: DATA  1,-1,0,-1,-1,-1,-1,
      0,-1,1,0,1,1,1,1,0
2990  RETURN
3000  :
3001  REM  *** PLAY
3002  :
3010  HOME : PRINT "3 2 1": PRINT
      "4 + 8": PRINT "5 6 7      W
      HIGH DIRECTION ==> "; CHR$
      (7);: GET AN$:ANS = VAL (A
      NS$): IF ANS < 1 OR ANS > 8 THEN
      3010
3015  HOME
3020  X2 = YX + DI(AN,1):Y2 = YY +
      DI(AN,2)
3025  COLOR= 0: PLOT YX + 9,YY +
      14: COLOR= 15: PLOT X2 + 9,Y
      2 + 14
3030  IF FI%(X2,Y2) = 2 THEN WL =
      0: RETURN : REM  FENCE
3031  IF FI%(X2,Y2) = 4 THEN WL =
      0: RETURN : REM  ROBOT
3032  IF FI%(X2,Y2) = 9 THEN WL =
      1: RETURN : REM  BASE
3035  FI%(YX,YY) = 0:YX = X2:YY =
      Y2:FI%(YX,YY) = 15
3040  FOR I = 1 TO NR
3045  IF  FN R(4) = 1 THEN X2 =  FN
      R(3) - 2:Y2 =  FN R(3) - 2: GOTO
      3055
3050  X2 =  SGN (YX - RO%(I,1)):Y2
      =  SGN (YY - RO%(I,2))
3055  X2 = X2 + RO%(I,1):Y2 = Y2 +
      RO%(I,2): IF FI%(X2,Y2) = 2 OR
      FI%(X2,Y2) = 4 OR FI%(X2,Y2)
      = 9 THEN 3045
3060  COLOR= 0: PLOT RO%(I,1) + 9
      ,RO%(I,2) + 14: COLOR= 4: PLOT
      X2 + 9,Y2 + 14
3065  IF FI%(X2,Y2) = 15 THEN WL =
      0: RETURN : REM  HUMAN

```



```

3070 FI%(RO%(I,1),RO%(I,2)) = 0:R
    O%(I,1) = X2:RO%(I,2) = Y2:F
    I%(RO%(I,1),RO%(I,2)) = 4
3075 NEXT I: GOTO 3010
3990 RETURN
4000 :
4001 REM *** END
4002 :
4010 HOME : PRINT "THE GAME IS O
    VER !!!"
4011 IF WL THEN PRINT "YOU'VE B
    EATEN THE KILLER ROBOTS (YEA
    H!)"
4012 IF NOT WL THEN PRINT "THE
    KILLER ROBOTS GOT YOU !!! (
    SORRY,.)"
4013 PRINT CHR$ (7); CHR$ (7); CHR$
    (7)
4990 RETURN

```



In this exciting game, you try to discern the combination to a safe. Your ears, as well as your eyes, are important tools. The object of the game is to open an enemy agent's safe before the thirty second delayed explosion kills you. Once you grasp all of the rules, you will discover that being a safe cracker is not too easy! Armed with a sophisticated safe cracking device, you try to detect the numbers in the combination one by one. Each time you pinpoint a number, you turn the paddle the other way until you pinpoint the next number in the combination. When you have identified the entire three-number combination, then the safe will open, and the explosion will be postponed. The three numbers in the combination are set by lines 2010, 2011, and 2012. You will note the word 'INVERSE' in line 2030. NORMAL sets the print mode to white letters on a black background. INVERSE reverses this so that you get black letter on a white background. Lines 2030 and 2035 draw the outline of the safe while in the INVERSE mode. That is how a white outline is drawn around the safe.

Our resident critic did not think that

<<< B O O M >>>

fulfilled the promise of a "terrible explosion". As a programmer trainee this is just the sort of routine you can manufacture to match your own expectations.

```

10 REM *****
11 REM *** ***
12 REM *** SAFE CRACKER ***
13 REM *** ***
14 REM *****
15 REM
16 REM
20 GOSUB 1000: REM INSTS
30 GOSUB 2000: REM SETUP
40 GOSUB 3000: REM PLAY!
50 GOSUB 4000: REM !END!
60 END

1000 :
1001 REM *** INSTS
1002 :
1010 TEXT : NORMAL : HOME
1020 VTAB 3: HTAB 10: PRINT "***
SAFE CRACKER ***"
1030 VTAB 7: PRINT "YOU ARE A GO
VERNMENT SPY, AND YOU MUST
RETRIEVE SOME CLASSIFIED DOC
UMENTS WHICH WERE STOLEN BY F
OREIGN AGENTS."
1040 PRINT : PRINT "THE DOCUMENT
S ARE KEPT IN A VAULT WHICH
YOU MUST OPEN."
1050 PRINT : PRINT "YOU HAVE BEE
N GIVEN A SOPHISTICATED SAFE
CRACKING DEVICE. WHENEVER T
HE TUMBLERS IN A SAFE CLICK
INTO PLACE, THE DEVICE WILL
ALSO MAKE A CLICKING SOUND.
IF YOU EITHER DIRECTLY HIT
OR PASS BY A NUMBER"
1052 PRINT "IN THE COMBINATION,
THEN THE DEVICE WILL EMIT A
CLICK."
1055 PRINT : PRINT
1060 INPUT "HIT RETURN WHEN READ
Y TO CONTINUE : ";ANS$
1070 HOME : VTAB 3: HTAB 10: PRINT
"*** SAFE CRACKER ***": VTAB
7

```

```

1080 PRINT "START BY TURNING PAD
DLE 0 ALL THE WAY TOTHE LEFT
(VALUE OF 0). THEN MOVE T
HE PADDLE TO THE RIGHT UNTI
L YOU GET THE FIRST NUMBER
, "
1090 PRINT : PRINT "WHEN YOU DIS
COVER THE FIRST NUMBER, THEN
TURN TO THE LEFT UNTIL YOU G
ET THE 2ND NUMBER, FINALLY
, TURN THE DIAL BACK TO THE
RIGHT FOR THE THIRD AND LAST
NUMBER. "
1095 PRINT : PRINT "IF YOU GO PA
ST A NUMBER, THEN YOU MUST
TURN THE DIAL ALL THE WAY TO
THE LEFT, AND START OVER."

1100 VTAB 23: INPUT "HIT RETURN
WHEN READY TO CONTINUE : ";A
NS$
1110 HOME : VTAB 3: HTAB 10: PRINT
"*** SAFE CRACKER ***": VTAB
7
1120 PRINT "OH, BY THE WAY, ONCE
YOU HAVE TOUCHED THE SAFE
, YOU WILL HAVE THIRTY SECON
DS TO OPEN IT. WHEN THIRTY
SECONDS HAS PASSED, THE
SAFETY MECHANISM WILL CAUSE
A TERRIBLE EXPLOSION."
1130 VTAB 23: INPUT "HIT RETURN
WHEN READY TO CONTINUE : ";A
NS$
1990 RETURN
2000 :
2001 REM *** SETUP
2002 :
2010 N1 = INT ( RND (1) * 60) +
1
2011 N2 = INT ( RND (1) * 60) +
1: IF N2 > = N1 THEN 2010

```



```

2012 N3 = INT ( RND (1) * 60) +
      1: IF N3 < = N2 THEN 2010
2020 HOME : VTAB 3: HTAB 10: PRINT
      "*** SAFE CRACKER ***"
2030 INVERSE : VTAB 5: HTAB 14: PRINT
      "          ": VTAB 15: HTAB
      14: PRINT "          "
2035 FOR I = 6 TO 14: VTAB I: HTAB
      14: PRINT " "; HTAB 25: PRINT
      " ";: NEXT NORMAL
2040 TI = 300:F = 1:D = 1
2045 V2 = 10:H2 = 19
2990 RETURN
3000
3001 REM *** PLAY
3002
3010 TI = TI - 1:T = INT (TI / 1
      0): VTAB 17: HTAB 15: PRINT
      "TIME :          ": VTAB 17: HTAB
      22: PRINT T + 1
3015 IF T + 1 = 0 THEN WL = 0: RETURN

3020 P = INT ( PDL (0) / 4.25)
3022 VTAB 7: HTAB 19: PRINT " "
      ;: VTAB 7: HTAB 19: PRINT P;

3023 GOSUB 3700
3024 IF P > 0 AND F THEN VTAB 9
      : HTAB 1: PRINT "TURN DIAL":
      VTAB 10: HTAB 1: PRINT "TO
      THE LEFT": GOTO 3010
3025 IF P = 0 AND F THEN F = 0: VTAB
      9: HTAB 1: PRINT " "
      ;: VTAB 10: HTAB 1: PRINT "
      ";
3030 VTAB 7: HTAB 19: PRINT " "
      ;: VTAB 7: HTAB 19: PRINT P;

3035 ON D GOTO 3040,3050,3060
3040 IF P < N1 THEN 3010
3041 IF P > N1 THEN GOSUB 3500:
      GOSUB 3600:F = 1: GOTO 3010

```

```

3042  GOSUB 3500:D = 2: GOTO 3010

3050  IF P > N1 THEN  GOSUB 3600:
      F = 1:D = 1: GOTO 3010
3054  IF P > N2 THEN 3010
3056  IF P < N2 THEN  GOSUB 3500:
      GOSUB 3600:F = 1: GOTO 3010

3058  GOSUB 3500:D = 3: GOTO 3010

3060  IF P < N2 THEN  GOSUB 3600:
      F = 1:D = 1: GOTO 3010
3064  IF P < N3 THEN 3010
3066  IF P > N3 THEN  GOSUB 3500:
      GOSUB 3600:F = 1: GOTO 3010

3068  GOSUB 3500:WL = 1: RETURN
3500  VTAB 5 + D: HTAB 30: PRINT
      "<CLICK>": FOR I = 1 TO 10:X
      X = PEEK ( - 16336): NEXT :
      RETURN
3600  VTAB 6: HTAB 30: PRINT "
      ": VTAB 7: HTAB 30: PRINT
      "      ": VTAB 8: HTAB 30: PRINT
      "      ": RETURN
3700  PP = P - INT (P / 4) * 4: ON
      PP + 1 GOTO 3701,3702,3703,3
      704
3701  V = 10:H = 19: GOTO 3705
3702  V = 11:H = 20: GOTO 3705
3703  V = 12:H = 19: GOTO 3705
3704  V = 11:H = 18: GOTO 3705
3705  VTAB V2: HTAB H2: PRINT " "
      ;: VTAB V: HTAB H: PRINT "*"
      ;:V2 = V:H2 = H: RETURN
3710  VTAB 23: HTAB 1: PRINT PP: RETURN
4000  :
4001  REM  *** END
4002  :
4010  IF WL = 1 THEN 4040

```

```

4015  HOME : INVERSE : FOR I = 1 TO
      10: VTAB I: PRINT "*****
      *****
      ***"; VTAB 21 - I: PRINT "*"
      *****
      *****"; NEXT I: NORMAL

4020  VTAB 10: HTAB 10: PRINT " <
      < < B O O M ! > > > ": FOR
      I = 1 TO 100:XX = PEEK ( -
      16336) + PEEK ( - 16336) -
      PEEK ( - 16336): NEXT
4030  VTAB 22: PRINT "THE COMBINA
      TION WAS : ";N1"- "N2"- "N3: RETURN

4040  VTAB 21: PRINT CHR$ (7); CHR$
      (7); CHR$ (7); "THE PAPERS AR
      E YOURS !!!": PRINT "YOUR CO
      LLEAGUES WILL BE QUITE IMPRE
      SSED.": RETURN

```



This is another two-player paddle game. Each player controls the up-and-down movements of a flying saucer. The object is to shoot your opponent's ship. The first player to do this three times is the winner. Line 3010 is responsible for, among other things, drawing the field of stars through which you must shoot. Lines 3520 and 3570 black out the prior position of each ship. To verify this, change COLOR= 0 to COLOR=1. Each time saucer #1 is moved, line 3605 draws the saucer in the new position. Line 3705 does the same for saucer #2. Line 3872 blacks out the previous position of each bomb that you shoot. Change COLOR= 0 to COLOR=8. If you manage to shoot your opponent, line 3950 draws the magenta (COLOR= 1) squares, and makes the corresponding noises (PEEK ( - 16336)).

```

10 REM *****
11 REM *** ***
12 REM *** SAUCER DUELS ***
13 REM *** ***
14 REM *****
15 REM
16 REM
20 GOSUB 1000: REM INSTS
30 GOSUB 2000: REM SETUP
40 GOSUB 3000: REM PLAY!
50 GOSUB 4000: REM !END!
60 END
1000 :
1001 REM *** INSTS
1002 :
1010 TEXT : NORMAL : HOME
1020 VTAB 3: HTAB 10: PRINT "***
    SAUCER DUELS ***"
1030 VTAB 7: PRINT "THIS IS A TW
    O-PLAYER GAME. EACH PLAYER
    CONTROLS A FLYING SAUCER BY
    USING THE PADDLES."
1035 PRINT
1040 PRINT "PADDLE 0 IS FOR PLAY
    ER #1, AND IS ON THE LEFT SID
    E OF THE SCREEN. PADDLE 1 I
    S FOR PLAYER #2, AND IS ON
    THE RIGHT SIDE OF THE SCREE
    N."
1045 PRINT
1050 PRINT "MOVE THE SAUCERS UP
    AND DOWN WITH THE PADDLE C
    ONTROL. TO SHOOT AT THE ENE
    MY SHIP, PRESS YOUR BUTTON.
    THE FIRST ONE TO SCORE THR
    EE HITS WINS THE GAME."
1060 VTAB 23: INPUT "HIT RETURN
    WHEN READY TO CONTINUE : ";AN
    S$
1990 RETURN
2000 :
2001 REM *** SETUP
2002 :

```

```

2010 P1 = 0:P2 = 0
2020 DEF FN R(X) = INT ( RND (
      1) * X)
2990 RETURN
3000 :
3001 REM *** PLAY
3002 :
3005 F1 = 0:F2 = 0:HI = 0:L2 = -
      1:L4 = - 1
3010 GR : HOME : FOR I = 1 TO 50
      :X = FN R(24) + 8:Y = FN R
      (40): COLOR= FN R(14): PLOT
      X,Y: NEXT
3020 PRINT "PLAYER #1
           "      "P1"
           "P2
3030 GOSUB 3500: REM MOVE SHIPS
3040 GOSUB 3800: REM MOVE SHOTS
3050 IF HI = 0 THEN 3030
3060 IF HI = 1 THEN P1 = P1 + 1:
      GOTO 3080
3070 IF HI = 2 THEN P2 = P2 + 1:
      GOTO 3080
3080 IF P1 < 3 AND P2 < 3 THEN 3
      005
3090 RETURN
3500 L1 = INT ( PDL (0) / 6.8):L
      3 = INT ( PDL (1) / 6.8)
3510 IF L2 = L1 THEN 3550
3520 IF L2 < L1 THEN COLOR= 0: GOSUB
      3600:L2 = L2 + 1: COLOR= 15:
      GOSUB 3600: GOTO 3550
3530 COLOR= 0: GOSUB 3600:L2 = L
      2 - 1: COLOR= 15: GOSUB 3600
      : GOTO 3550
3550 IF L4 = L3 THEN 3590
3560 IF L4 < L3 THEN COLOR= 0: GOSUB
      3700:L4 = L4 + 1: COLOR= 15:
      GOSUB 3700: GOTO 3590

```

```

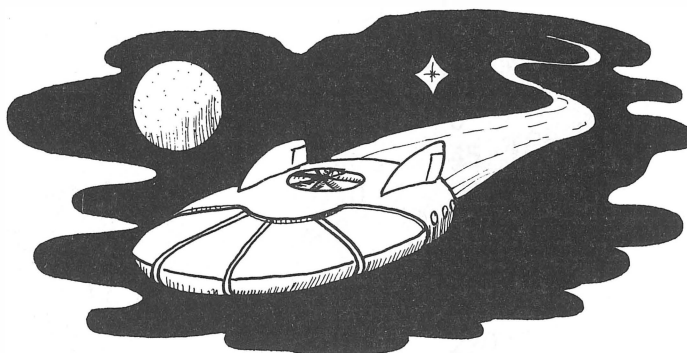
3570  COLOR= 0: GOSUB 3700:L4 = L
      4 - 1: COLOR= 15: GOSUB 3700
      : GOTO 3590
3590  RETURN
3600  IF L2 < 0 THEN RETURN
3605  HLIN 2,4 AT L2: HLIN 0,2 AT
      L2 + 1: HLIN 4,6 AT L2 + 1: HLIN
      1,5 AT L2 + 2: RETURN
3700  IF L4 < 0 THEN RETURN
3705  HLIN 35,37 AT L4: HLIN 33,3
      5 AT L4 + 1: HLIN 37,39 AT L
      4 + 1: HLIN 34,38 AT L4 + 2:
      RETURN
3800  IF F1 THEN 3850
3805  IF PEEK ( - 16287) < 128 THEN
      3850
3810  F1 = 1:X1 = 7:Y1 = L2 + 1: COLOR=
      12: PLOT X1,Y1
3850  IF F2 THEN 3870
3855  IF PEEK ( - 16286) < 128 THEN
      3870
3860  F2 = 1:X2 = 32:Y2 = L4 + 1: COLOR=
      12: PLOT X2,Y2
3870  FOR I = 1 TO 5
3871  IF NOT F1 THEN 3880
3872  COLOR= 0: PLOT X1,Y1
3873  X1 = X1 + 1: IF X1 > 39 THEN
      F1 = 0: GOTO 3880
3875  IF SCRN( X1,Y1) = 0 THEN COLOR=
      12: PLOT X1,Y1: GOTO 3880
3876  IF SCRN( X1,Y1) < > 15 THEN
      COLOR= 0: PLOT X1,Y1: GOSUB
      3900:F1 = 0: GOTO 3880
3878  GOSUB 3950:HI = 1: RETURN
3880  IF NOT F2 THEN 3890
3882  COLOR= 0: PLOT X2,Y2
3883  X2 = X2 - 1: IF X2 < 0 THEN
      F2 = 0: GOTO 3890
3885  IF SCRN( X2,Y2) = 0 THEN COLOR=
      12: PLOT X2,Y2: GOTO 3890
3886  IF SCRN( X2,Y2) < > 15 THEN
      COLOR= 0: PLOT X2,Y2: GOSUB
      3900:F2 = 0: GOTO 3890

```

```

3888  GOSUB 3960:HI = 2: RETURN
3890  NEXT : RETURN
3900  FOR J = 1 TO 5:XX = PEEK (
      - 16336): NEXT : RETURN
3950  COLOR= 1: FOR I = 1 TO 12: PLOT
      X1 - 6 + FN R(7),Y1 - 1 + FN
      R(3):XX = PEEK ( - 16336) -
      PEEK ( - 16336) + PEEK ( -
      16336) - PEEK ( - 16336): NEXT
      : RETURN
3960  COLOR= 1: FOR I = 1 TO 12: PLOT
      X2 + FN R(7),Y2 - 1 + FN R
      (3):XX = PEEK ( - 16336) -
      PEEK ( - 16336) + PEEK ( -
      16336) - PEEK ( - 16336): NEXT
      : RETURN
4000  :
4001  REM *** END
4002  :
4010  HOME : PRINT "THE GAME IS O
      VER !!!"
4020  IF P1 = 3 THEN PRINT "PLAY
      ER NUMBER 1 IS THE WINNER !!
      !"
4021  IF P2 = 3 THEN PRINT "PLAY
      ER NUMBER 2 IS THE WINNER !!
      !"
4990  RETURN

```





# Schmoo

This whimsical name and the humorous object of the game serve to camouflage an excellent thinking-man's game. It is hard to imagine how a person could play, and not come away with a better understanding of the X,Y coordinate system. The premise is that you are trying to splat a mudball on the mud-loving Schmoo. The elevation at which you aim the automatic mudball slinger determines how far the mudball will travel. The angle at which you shoot will be determined by the coordinates of the Schmoo. Following will be a list of coordinates and the angles they represent.

X	Y	ANGLE
12239	0	0
17866	17866	45
0	23910	90
-5888	5888	135
-9400	0	180
-25727	-25727	225
0	-18992	270
31101	-31101	315

We hope this chart will help you to understand how the various coordinates relate to the angles.

```

10 REM *****
11 REM *** ***
12 REM *** SCHMOO ***
13 REM *** ***
14 REM *****
15 REM
16 REM
20 GOSUB 1000: REM INSTS
30 GOSUB 2000: REM SETUP
40 GOSUB 3000: REM PLAY!
50 GOSUB 4000: REM !END!
60 END

1000 :
1001 REM *** INSTS
1002 :
1010 TEXT : HOME : NORMAL
1020 VTAB 1: HTAB 13: PRINT "***
      SCHMOO ***"
1030 VTAB 5: PRINT "THIS IS THE
      GAME OF SCHMOO. IN IT YOU
      THROW MUD AROUND IN HOPES OF
      HITTING THEMUD-LOVING SCHMO
      O."
1040 PRINT : PRINT "YOU ARE SITU
      ATED IN THE CENTER OF AN X,Y
      COORDINATE SYSTEM; AT POSITI
      ON 0,0. THESCHMOO WILL BE L
      OCATED SOMEWHERE ON THE SAME
      PLANE. HIS COORDINATES ARE
      GIVEN TO YOU BEFORE EACH T
      URN."
1045 VTAB 23: INPUT "HIT RETURN
      WHEN READY TO CONTINUE : ";A
      NS$
1047 HOME : VTAB 1: HTAB 13: PRINT
      "*** SCHMOO ***": VTAB 5
1050 PRINT : PRINT "YOU HAVE YOU
      R TRUSTY AUTOMATIC MUDBALL
      SLINGER WHICH YOU USE TO TOS
      S MUDBALLS AT THE SCHMOO.
      YOU INPUT THE ELEVATION AND
      THE ANGLE AT WHICH YOU WISH
      TO FIRE"

```



```

1055 PRINT "THE MUDBALL. AFTER
      EACH SHOT YOU WILL BE GIVEN
      THE COORDINATES WHERE THE M
      UD LANDED. "
1060 PRINT : PRINT "FOR EXAMPLE,
      IF THE SCHMOO'S COORDINATES
      ARE (-5,10) THEN THE SCHMOO
      IS ABOUT FIVE FEET TO YOU
      R LEFT AND ABOUT TEN FEET
      IN FRONT OF YOU."
1070 PRINT : PRINT "THE ELEVATIO
      N FOR THE SHOT WOULD BE
      ABOUT 89.95 DEGREES WHILE TH
      E ANGLE WHERE THE SCHMOO
      CAN BE FOUND IS ABOUT 110
      DEGREES."
1080 PRINT : INPUT "HIT RETURN W
      HEN READY TO CONTINUE : ";AN
      S$
1090 HOME : VTAB 1: HTAB 13: PRINT
      "*** SCHMOO ***": VTAB 5
1100 PRINT "THE MUDBALLS ARE LAR
      GE ENOUGH TO MUDDY THE SCHM
      OO AS LONG AS THEY LAND WITH
      IN 100 FEET OF HIM."
1110 PRINT : PRINT : PRINT "NOW
      THAT YOU KNOW HOW TO MAKE TH
      E SCHMOOHAPPY, GO GET HIM.
      GOOD LUCK!"
1120 VTAB 23: INPUT "HIT RETURN
      WHEN READY TO CONTINUE : ";A
      NS$
1990 RETURN
2000 :
2001 REM *** SETUP
2002 :
2010 S1 = INT ( RND (1) * 2) * 2
      - 1:S2 = INT ( RND (1) * 2
      ) * 2 - 1
2020 SX = ( INT ( RND (1) * 26000
      ) + 5000) * S1:SY = ( INT ( RND
      (1) * 26000) + 5000) * S1
2030 BELL$ = CHR$ (7)
2035 CNSTR = 3.1415926357989 / 18
      0

```

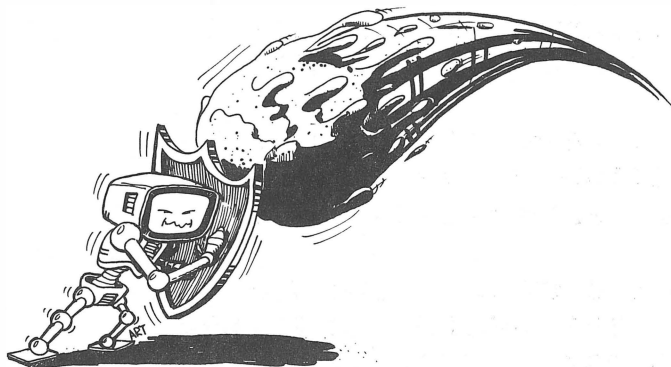
```

2990 RETURN
3000 :
3001 REM *** PLAY
3002 :
3005 HOME : VTAB 3: HTAB 13: PRINT
    "*** SCHMOO ***": VTAB 7
3010 PRINT : PRINT BELL$"THE SCH
    MOO IS AT COORDINATES : ": PRINT
    ("SX","SY")"
3015 PRINT : PRINT "WHAT ELEVATI
    ON FOR THE": INPUT "MUDBALL
    SLINGER (0-90) : ";EL
3020 IF EL > 90 OR EL < 1 THEN PRINT
    "THE ELEVATION RANGES FROM 1
    TO 90...": GOTO 3015
3025 IF EL = 90 THEN PRINT "THA
    T WOULD SHOOT THE MUD STRAIG
    HT UP....AND IT WOULD COME D
    OWN ON TOP OF YOU!": GOTO 30
    15
3030 PRINT : PRINT "WHAT ANGLE O
    F DIRECTION FOR THE": INPUT
    "MUDBALL SLINGER (0-360) : "
    ;AN
3035 IF AN < 0 OR AN > 360 THEN
    PRINT "THE ANGLES RANGES FR
    OM 0 TO 360.": GOTO 3030
3040 DM = ABS ( INT (93000 * SIN
    (EL * CN) * COS (EL * CN)))
3045 XM = DM * COS (AN * CN):YM =
    DM * SIN (AN * CN)
3050 DS = SQR ((SX - XM) ^ 2 + (
    SY - YM) ^ 2)
3055 PRINT : PRINT "THE MUD SPLA
    TTERED AT COORDIATES :": PRINT
    "(" INT (XM)"," INT (YM)")"
3060 TRY = TRY + 1
3065 IF DS < = 100 THEN PRINT
    : PRINT BELL$BELL$"THAT'S GO
    OD ENOUGH TO": PRINT "SPLAT
    THE SCHMOO !!!": RETURN
3070 GOTO 3010
4000 :

```

```
4001 REM *** END
4002 :
4010 PRINT : PRINT "YOU SPLATTED
      THE SCHMOO IN "TRY" TRIES."

4020 INPUT "DO YOU WISH TO PLAY
      AGAIN?";ANS$
4030 IF LEFT$(ANS$,1) = "Y" THEN
      RUN
4035 PRINT : PRINT "THANKS FOR S
      PLATTING THE SCHMOO !!"
4990 RETURN
```





True to its name, the object is to dodge the stars for as long as possible. Don't try to intercept the stars. If you do, you lose! This program is run entirely in the TEXT mode. This is done to aid your understanding of the graphics by using text format commands in place of graphic commands. You may notice the absence of certain commands, such as: HLIN, PLOT COLOR, GR, and others. Line 3055 uses an HTAB and a VTAB and a PRINT statement instead of: PLOT (variable name) AT SX,SY.

What do those POKES do in lines 2100 and 2110 do? If you add 2099 GOTO 2990 to isolate these lines from the program you will be surprised. The program runs just the same! But does it? If you already ran the program the POKES put a machine language routine in memory that stays there even when you type RUN, NEW, or even PR#6. Turn the computer off and on. Load Stardodger (don't run it). Type 2099 GOTO 2990 and run the program. Now you see that these lines created the sounds that accompanied the stars as they popped onto the screen. The sounds are CALL(ed) in line 3035. You can make your own music with this machine language routine and add sounds to your own programs.

DEL(ete) all of the program except lines 2100 and 2110

ADD:

```
2120 X = PDL(0):Y = PDL(1)
2130 POKE 0,X :POKE 1,Y
2140 CALL 768
2150 GOTO 2120
```

You will have an instrument that plays background music for low budget science fiction movies.

Another variation on that theme:

```
2120 GET A$
2130 X = ASC (A$)
2140 POKE 0,X:POKE 1,50
2150 CALL 768
2160 GOTO 2120
```

Turns the Apple keyboard into a piano; even the control keys and RETURN play music.

```
10 REM *****
11 REM *** ***
12 REM *** STARDODGER ***
13 REM *** ***
14 REM *****
15 REM
16 REM
20 GOSUB 1000: REM INSTS
30 GOSUB 2000: REM SETUP
40 GOSUB 3000: REM PLAY!
50 GOSUB 4000: REM !END!
60 END
1000 :
1001 REM *** INSTS
1002 :
1010 TEXT : NORMAL : HOME
1020 VTAB 3: HTAB 11: PRINT "***
    STARDODGER ***"
1030 VTAB 5: PRINT "IN THIS GAME
    , YOU WILL BE USING THE GAME
    PADDLE TO PILOT A SPACESHIP.
    THE OBJECT IS TO MANEUVER T
    HE SHIP THROUGH A FIELD OF S
    TARS, AND TO SUCCESSFULLY RE
    ACH THE BOTTOM OF THE SCREEN
    ,"
```

```

1035 PRINT : PRINT "YOU MUST TRY
      TO AVOID COLLIDING WITH THE
      STARS. EACH COLLISION WITH
      THE STARS DRAINS POWER FRO
      M YOUR SHIP'S PROTECTIVESHIE
      LDS."
1040 PRINT : PRINT "YOU HAVE FIV
      E ATTEMPTS TO REACH THE HOME
      BASE BEFORE YOUR SHIELDS GIV
      E OUT. AT THIS POINT, THE
      NEXT COLLISION WILL FINDYOUR
      SHIP WITH NO POWER AND THE
      SHIP WILL BE DESTROYED >
      BOOM! <"
1043 PRINT
1045 CALL - 958: PRINT "HOW DO
      YOU RATE YOURSELF ?": INPUT "
      1=POOR ... 10=GREAT. ";ANS:
      IF ANS < 1 OR ANS > 10 OR A
      NS < > INT (ANS) THEN 1045

1990 RETURN
2000 :
2001 REM *** SETUP
2002 :
2010 DEF FN R(X) = INT ( RND (
      1) * X) + 1
2015 NT = 5
2020 DUR = (11 - ANS) * 5
2025 BE$ = CHR$ (7)
2100 POKE 768,173: POKE 769,48: POKE
      770,192: POKE 771,136: POKE
      772,208: POKE 773,4: POKE 77
      4,198: POKE 775,1: POKE 776,
      240
2110 POKE 777,8: POKE 778,202: POKE
      779,208: POKE 780,246: POKE
      781,166: POKE 782,0: POKE 78
      3,76: POKE 784,0: POKE 785,3
      : POKE 786,96
2990 RETURN
3000 :
3001 REM *** PLAY

```

```

3002 :
3010 HOME : VTAB 10: IF NT > 1 THEN
    PRINT NT" ATTEMPTS REMAIN !
    !!": GOTO 3014
3012 PRINT "1 ATTEMPT REMAINS !!
    !"
3014 FOR I = 1 TO 1000: NEXT I
3015 SX = FN R(38):SY = 1: HOME

3017 X2 = SX:Y2 = SY
3020 VTAB SY: HTAB SX: PRINT "
    ";SX = X2:SY = Y2
3025 FOR I = 1 TO ANS / 3: VTAB
    24: HTAB FN R(40): PRINT MID$
    ("*+X", FN R(4),1);: NEXT I
    : VTAB 24: HTAB 40: PRINT
3035 POKE 0, FN R(50) + 200: POKE
    1,DUR: CALL 768
3040 IF SCRN( SX - 1,SY * 2) <
    > 0 OR SCRN( SX,SY * 2) <
    > 0 OR SCRN( SX + 1,SY * 2
    ) < > 0 THEN 3100
3045 IF PDL (0) < 20 THEN X2 =
    SX - 2: GOTO 3050
3046 IF PDL (0) < 90 THEN X2 =
    SX - 1: GOTO 3050
3047 IF PDL (0) > 165 THEN X2 =
    SX + 1: GOTO 3050
3048 IF PDL (0) > 235 THEN X2 =
    SX + 2: GOTO 3050
3050 IF X2 < 1 THEN X2 = 1
3051 IF X2 > 38 THEN X2 = 38
3055 VTAB SY: HTAB SX: PRINT "<*
    >";Y2 = SY
3060 CNT = CNT + 1: IF CNT = 8 THEN
    CNT = 0:Y2 = Y2 + 1: IF Y2 =
    20 THEN 3200
3065 FOR I = 1 TO 50: NEXT
3070 GOTO 3020
3100 SX = SX - 2: IF SX < 1 THEN
    SX = 1
3110 SY = SY - 2: IF SY < 1 THEN
    SY = 1

```



```

3120  FOR I = 1 TO 5: FOR J = SY TO
      SY + 4: HTAB X2 + I - 1: VTAB
      J
3130  PRINT MID$ (".....,---
      --+++++XXXXX",(I - 1) * 5 +
      1,5);: NEXT J: PRINT BE$;: NEXT
      I
3160  NT = NT - 1: IF NT > 0 THEN
      3010
3200  RETURN
4000  :
4001  REM  *** END
4002  :
4010  VTAB 21: HTAB 1: CALL  - 95
      8: VTAB 22
4020  IF NT = 0 THEN PRINT BE$BE
      $BE$"I'M SORRY, BUT YOU LOST
      ...": RETURN
4030  PRINT BE$BE$BE$"YOU WON !!!
      CONGRATULATIONS !!!": RETURN

4990  RETURN

```





Grab a friend and get ready for some heated competition. The object of this game is to wall-in your opponent and to prevent him from moving. On each turn, you enter the coordinates of the adjacent square you would like to move into, plus, you enter the coordinates of a square you wish to become uninhabitable. The first player unable to move loses the game. Line 2030 switches the mode to G`R`aphics. Next, the color is set to a yellowish-green (`COLOR = 12`). Line 2031 draws a Horizontal `LINE` starting at position zero and continuing to position thirty-two. The result is a solid yellowish-green block, 32 by 32. Line 2033 sets the color to 15 (white), and then begins a loop (consisting of nine passes). Line 2034 draws a white Horizontal `LINE` on top of every fourth green line. This line also instructs the computer to draw a Vertical `LINE` at each value of "I". The result is an 8 by 8 matrix, comprised of 64 green squares, each with a white border. Lines 2040 and 2041 (and 5001) draw the magenta (`COLOR = 1`) and the light green (`COLOR = 14`) squares which mark each player's beginning position. To verify this, type: 2041 `STOP` (then return). Now when you run the program, it will `STOP` at line 2041. Only the square drawn by line 2040 be displayed. To undo any change, type: `LOAD STRANDED`. When this is done, a copy of the old, unaltered program is moved from permanent storage on diskette into the computer's memory, where it can be modified and/or run.

```

10 REM *****
11 REM *** ***
12 REM *** STRANDED ***
13 REM *** ***
14 REM *****
15 REM
16 REM
20 GOSUB 1000: REM INSTS
30 GOSUB 2000: REM SETUP
40 GOSUB 3000: REM PLAY!
50 GOSUB 4000: REM !END!
60 END
1000 :
1001 REM *** INSTS
1002 :
1010 TEXT : NORMAL : HOME
1020 VTAB 3: HTAB 12: PRINT "***
STRANDED ***"
1030 VTAB 7: PRINT "THIS IS A GA
ME FOR TWO PLAYERS. BOTH OF
YOU WILL BE PLACED IN AN 8*8
MATRIX. A PLAYER MAY MOV
E IN ANY OF THE EIGHT DIRE
CTIONS. ENTER THE COORDINAT
ES OF THE TARGET SQUARE. "

1035 PRINT
1040 PRINT "AFTER YOU MOVE, YOU
WILL BE ASKED FOR THE COOR
DINATES OF A SECOND SQUARE.
THISSQUARE WILL THEN BE BLAC
KED OUT, AND BE UNENTERABLE.
"

3040 IF SQR (((X1 - X) ^ 2) + (
(Y1 - Y) ^ 2)) > = 2 THEN PRINT
CHR$ (7): GOTO 3020
3050 IF M(X1,Y1) < > 0 THEN PRINT
CHR$ (7): GOTO 3020
3060 M(X,Y) = 0:CO = 12: GOSUB 50
00
3070 M(X1,Y1) = PL:CO = FN P(PL)
:P(PL,PL) = X1:P(PL,PL + 1) =
Y1:X = X1:Y = Y1: GOSUB 5000

```

```

3100  INPUT "BLOCK AT : ";X2,Y2: IF
      X2 < 1 OR X2 > 8 OR Y2 < 1 OR
      Y2 > 8 OR X2 < > INT (X2) OR
      Y2 < > INT (Y2) THEN 3100
3110  IF M(X2,Y2) < > 0 THEN 310
      0
3120  M(X2,Y2) = 3:CO = 0:X = X2:Y
      = Y2: GOSUB 5000
3200  K = 3 - PL: FOR J = 1 TO 8:X
      4 = DIR(J,1) + P(K,K):Y4 = D
      IR(J,2) + P(K,K + 1)
3210  IF X4 < 1 OR X4 > 8 OR Y4 <
      1 OR Y4 > 8 THEN 3250
3220  IF M(X4,Y4) < > 0 THEN 325
      0
3230  J = 8: NEXT J: GOTO 3990
3250  NEXT J: RETURN
3990  NEXT
3995  GOTO 3010
4000  :
4001  REM *** END
4002  :
4010  HOME : PRINT "THE GAME IS O
      VER.....": PRINT "PLAYER NUM
      BER "PL" IS THE WINNER !"
4020  RETURN
5000  REM *** DRAW A SQUARE
5001  COLOR= CO: FOR I = FN C(X)
      TO FN C(X) + 2: HLIN FN C
      (Y), FN C(Y) + 2 AT I: NEXT
      : RETURN

```



This game requires good timing. A small, moving target is the object of your marksman talent. The angle of each shot depends on the angle of the pad. Unlike most of the games where the game board is drawn in the the '2000' subroutine, here, all of the GGraphics are drawn in the '3000' subroutine. Looking at the graphics, line 3010 draws a light blue (COLOR = 7) perimeter around the target area. This can be verified by typing: 3012 STOP. Now when you run the program, the execution will STOP immediately following the completion of 3010. All that will be on the screen is the blue outline. Line 3030 draws the launching pad in a symmetrical configuration, but at a RaNDom location and utilizing a RaNDom five-dot design. Again, to confirm the function of line 3030, type: 3029 STOP and 3031 STOP. When the program encounters a STOP command, you can CONTInue the execution (the run) by typing CONT. Line 3040 draws the pink (COLOR = 11) target at its original position and each time it moves down the screen. Also, note that before the color equals eleven, the color is set to zero (black). To understand the function of the first part of line 3030, change COLOR = 0 to COLOR = 1. The function of line 3075 is similar to that of line 3030. Experiment with this group of instructions to see if you can discover 3075's purpose.

```

10 REM *****
11 REM *** ***
12 REM *** TARGET ***
13 REM *** ***
14 REM *****
15 REM
16 REM
20 GOSUB 1000: REM INSTS
40 GOSUB 3000: REM PLAY!
60 END
1000 :
1001 REM *** INSTS
1002 :
1010 TEXT : NORMAL : HOME
1020 VTAB 3: HTAB 13: PRINT "***
    TARGET ***"
1030 VTAB 7: PRINT "IN THIS GAME
    YOU TRY TO HIT A MOVING
    TARGET. BY PRESSING ANY KEY
    , A SMALL BALL WILL BE FIR
    ED FROM THE PADDLE ON THE
    SCREEN."
1040 PRINT : PRINT "YOU MUST TIM
    E THE RELEASE SUCH THAT THE
    SMALL BALL HITS THE LARGER O
    NE. THE DIRECTION AND DI
    STANCE WILL VARY WITH EACH
    NEW TARGET."
1090 VTAB 23: INPUT "HIT RETURN
    WHEN READY TO CONTINUE : ";A
    NS$
1990 RETURN
3000 :
3001 REM *** PLAY
3002 :
3010 GR : HOME : COLOR= 7: HLIN
    0,39 AT 0: HLIN 0,39 AT 39: VLIN
    0,39 AT 0: VLIN 0,39 AT 39
3015 HOME : VTAB 22: HTAB 10: PRINT
    "SCORE: ";SC;" SHOTS: ";S
    H
3020 IF SH THEN VTAB 23: HTAB 1
    4: PRINT "PCENT: "; INT (SC /
    SH * 100);"%

```



```

3025 PM = INT ( RND (1) * 2) * 2
      - 1:XP = INT ( RND (1) * 1
0) + 10:YP = INT ( RND (1) *
10) + 15:SP = INT ( RND (1)
* 3) * PM:FLAG = 0
3030 COLOR= 9: PLOT XP + (2 * SP
),YP - 2: PLOT XP + SP,YP -
1: PLOT XP,YP: PLOT XP - SP,
YP + 1: PLOT XP - (2 * SP),Y
P + 2
3035 X2 = INT ( RND (1) * 10) +
25:Y2 = 1:S2 = INT ( RND (1
) * 2) + 1:X3 = X2:Y3 = Y2
3040 COLOR= 0: FOR I = X3 - 1 TO
X3 + 1: VLIN Y3,Y3 + 2 AT I:
NEXT : COLOR= 11: FOR I = X
2 - 1 TO X2 + 1: VLIN Y2,Y2 +
2 AT I: NEXT
3045 X3 = X2:Y3 = Y2: IF FL THEN
3070
3050 IF PEEK ( - 16384) < 127 THEN
3600
3055 POKE - 16368,0:FL = 1:B1 =
XP + 1:B2 = YP:B3 = B1:B4 =
B2:SH = SH + 1
3060 HOME : VTAB 22: HTAB 10: PRINT
"SCORE: ";SC;" SHOTS: ";S
H
3065 VTAB 23: HTAB 14: PRINT "PC
ENT: "; INT (SC / SH * 100);
"%": IF B1 < > B3 AND SCRN(
B3,B4) < > 15 THEN 3095
3070 IF SCRN( B3,B4) = 11 THEN
3095
3075 COLOR= 0: PLOT B3,B4: COLOR=
15: PLOT B1,B2:B3 = B1:B4 =
B2:B1 = B1 + 1:B2 = B2 + SP
3080 IF B2 < 1 OR B2 > 38 THEN B
2 = B4:SP = - SP: GOTO 3600

3085 IF B1 > 36 THEN COLOR= 0: PLOT
B3,B4:FLAG = 0:SP = ABS (SP
) * PM: GOTO 3600

```

```
3090 IF SCRN( B1,B2) = 0 THEN 3
      600
3095 SC = SC + 1: GOTO 3010
3600 Y2 = Y2 + 1: IF Y2 > 36 THEN
      Y2 = 1
3610 GOTO 3040
3990 RETURN
```

In this game the object is to avoid the relentless pursuit of the horrible Twinky, and to escape from the danger-filled labyrinth. There are a plethora of obstacles which impede your escape. There are twenty squares which relocate you somewhere in the maze. There are twenty squares which cannot be entered. There is one square which contains an extremely sensitive exploding device. If you move onto this space, the ensuing blast will end your perilous journey . . . and your life.

```

10 REM *****
11 REM *** ***
12 REM *** TWINKY ***
13 REM *** ***
14 REM *****
15 REM
16 REM
20 GOSUB 1000: REM INSTS
30 GOSUB 2000: REM SETUP
40 GOSUB 3000: REM PLAY!
60 END
1000 :
1001 REM *** INSTS
1002 :
1010 TEXT : HOME : NORMAL
1020 VTAB 2: HTAB 13: PRINT "***
TWINKY ***"
1030 VTAB 5: PRINT "THIS IS THE
GAME OF TWINKY. IN IT YOU
PRETEND TO BE A SPACE EXPLOR
ER WHO HAS LANDED ON A HOST
ILE PLANET. "
1040 PRINT : PRINT "CAPTURED BY
THE UNFRIENDLY NATIVES, YOU
ARE TOSSED INTO A LARGE PRIS
ON ALONG WITH A FEROCIOUS
TWINKY."
1050 PRINT : PRINT "A TWINKY IS
A HORRIBLE CREATURE THAT
WILL CATCH YOU AND ABSORB YO
UR BODY INTO HIS IF HE GETS C
LOSER THAN TWO UNITS AWAY
FROM YOU."
1060 VTAB 23: INPUT "HIT RETURN
WHEN READY TO CONTINUE :";AN
S$
1070 HOME : VTAB 2: HTAB 13: PRINT
"*** TWINKY ***": VTAB 5
1080 PRINT "IN THE INTEREST OF F
AIR PLAY, YOU ARE GIVEN A
ZAP GUN THAT WILL TEMPORARIL
Y CHASE THE TWINKY AWAY."

```

```

1090 PRINT : PRINT "ALSO, IF YOU
      CAN MAKE IT TO THE SPECIAL
      OBJECTIVE SQUARE BEFORE BEIN
      G ABSORBED, YOU WILL BE SET
      FREE."
1100 PRINT : PRINT "AFTER YOU MO
      VE, YOU WILL BE INFORMED OF
      YOUR DISTANCE FROM THE OBJEC
      TIVE SQUARE AS WELL AS FROM
      THE TWINKY.  "
1110 VTAB 23: INPUT "HIT RETURN
      WHEN READY TO CONTINUE :";AN
      S$
1120 HOME : VTAB 2: HTAB 13: PRINT
      "*** TWINKY ***": VTAB 5
1130 PRINT "THERE ARE SEVERAL OT
      HER OBJECTS WITHIN THE MAZE
      WHICH ARE OF INTEREST."
1140 PRINT : PRINT "THERE ARE TW
      ENTY RELOCATION SQUARES.
      THESE SQUARES SEND YOU TO SO
      ME OTHER SECTION OF THE M
      AZE.  "
1150 PRINT : PRINT "THERE ARE TW
      ENTY IMPREGNABLE SQUARES
      WHICH YOU CANNOT ENTER."
1160 PRINT : PRINT "THERE IS ONE
      SUPER DEADLY AUTOMATIC KILL
      SQUARE WHICH ENDS YOUR ORDEA
      L QUICKLY AND PAINLESSLY."

1170 PRINT : PRINT : PRINT "THAT
      'S IT... TRY TO ENJOY IT !"
1180 VTAB 23: INPUT "HIT RETURN
      WHEN READY TO CONTINUE :";AN
      S$
1990 RETURN
2000 :
2001 REM *** SETUP
2002 :
2010 DIM MA(15,15)
2015 DEF FN R(X) = INT ( RND (
      1) * X) + 1: DEF FN A(X) =
      .001 * INT (X * 1000 + .5)

```

```

2020  FOR I = 1 TO 20
2025  X = FN R(15):Y = FN R(15):
      IF MA(X,Y) THEN 2025
2030  MA(X,Y) = 1: NEXT I: REM  **
      * BLOCKED
2040  FOR I = 1 TO 20
2045  X = FN R(15):Y = FN R(15):
      IF MA(X,Y) THEN 2045
2050  MA(X,Y) = 2: NEXT I: REM  **
      * RELOCATION
2065  X = FN R(15):Y = FN R(15):
      IF MA(X,Y) THEN 2065
2070  MA(X,Y) = 3: REM  *** SUPER
      KILL
2075  XO = FN R(15):YO = FN R(15
      ): IF MA(XO,YO) THEN 2075
2080  MA(XO,YO) = 4: REM  *** OBJE
      CTIVE
2085  XT = FN R(15):YT = FN R(15
      ): IF MA(XT,YT) THEN 2085
2090  MA(XT,YT) = 5: REM  *** TWIN
      KY
2095  XP = FN R(15):YP = FN R(15
      ): IF MA(XP,YP) THEN 2095
2100  MA(XP,YP) = 6: REM  *** PLAY
      ER
2110  ST = 0:SP = 0
2990  RETURN
3000  :
3001  REM  *** PLAY
3002  :
3010  HOME : VTAB 3: HTAB 13: PRINT
      "*** TWINKY ***": PRINT : PRINT
      : PRINT
3020  DT = FN A( SQR ((XT - XP) ^
      2 + (YT - YP) ^ 2))
3021  DO = FN A( SQR ((XP - XO) ^
      2 + (YP - YO) ^ 2))
3025  PRINT : PRINT "THE TWINKY I
      S "DT" UNITS AWAY"
3026  PRINT : PRINT "THE OBJECTIV
      E IS "DO" UNITS AWAY"

```

```

3028 IF DT < 2 THEN : PRINT : PRINT
    "《《《《 S C H L O O R P ! ! !
    >>>>>": PRINT "YOU'VE BEEN
    ABSORBED BY THE TWINKY !!!":
    PRINT "YOU LOSE,":WL = 1: GOSUB
    3600: RETURN
3030 PRINT : INPUT "MOVE OR SHOO
    T (M/S) : ";ANS$
3035 ANS$ = LEFT$(ANS$,1): IF A
    NS$ < > "M" AND ANS$ < > "
    S" THEN PRINT "TYPE IN 'M'
    OR 'S'": GOTO 3030
3040 IF ANS$ = "S" THEN 3300
3100 PRINT : PRINT "FORWARD, BAC
    KWARD,": INPUT "RIGHT OR LEF
    T (F/B/R/L) : ";ANS$
3105 ANS$ = LEFT$(ANS$,1): IF A
    NS$ < > "F" AND ANS$ < > "
    B" AND ANS$ < > "R" AND ANS
    $ < > "L" THEN PRINT "TYPE
    IN 'F' OR 'B' OR 'R' OR 'L'
    ": GOTO 3100
3110 IF ANS$ = "F" THEN X = 0:Y =
    - 1: GOTO 3120
3111 IF ANS$ = "B" THEN X = 0:Y =
    1: GOTO 3120
3112 IF ANS$ = "R" THEN X = 1:Y =
    0: GOTO 3120
3113 IF ANS$ = "L" THEN X = - 1
    :Y = 0: GOTO 3120
3120 X = X + XP:Y = Y + YP
3125 IF X < 1 OR X > 15 OR Y < 1
    OR Y > 15 THEN PRINT "THAT
    WOULD TAKE YOU OUT OF THE M
    AZE": PRINT "MOVE NOT ALLOWE
    D": GOTO 3500
3130 IF MA(X,Y) = 1 THEN PRINT
    "THAT SPACE IS BLOCKED": PRINT
    "MOVE NOT ALLOWED": GOTO 350
    0
3135 IF MA(X,Y) = 2 THEN 3200

```

```

3140 IF MA(X,Y) = 3 THEN PRINT
"YOU FOUND THE SUPER KILL SQ
UARE!!!": PRINT "MOVE ALLOWE
D BUT,": PRINT "YOU'VE BEEN
KILLED!!!":WL = 1: RETURN
3145 IF MA(X,Y) = 4 THEN PRINT
"YOU FOUND THE OBJECTIVE !!!
": PRINT "MOVE ALLOWED AND,"
: PRINT "YOU WIN A TRIP OFF
THIS PLANET !!!":WL = 0: RETURN

3150 IF MA(X,Y) = 5 THEN PRINT
"MOVE ALLOWED": PRINT : PRINT
: PRINT "<<<<< S C H L O O R
P ! ! ! >>>>>": PRINT "YOU'
VE BEEN ABSORBED BY THE TWIN
KY !!!": PRINT "YOU LOSE.":W
L = 1: GOSUB 3600: RETURN
3155 PRINT "MOVE ALLOWED":MA(XP,
YP) = SP:XP = X:YP = Y:SP =
MA(XP,YP):MA(XP,YP) = 6: GOTO
3500
3200 PRINT "..... YOU'VE BEEN RE
LOCATED ....."
3205 X = FN R(15):Y = FN R(15)
3210 IF MA(X,Y) = 1 THEN 3205
3215 GOTO 3135
3300 PRINT : PRINT "FORWARD, BAC
KWARD,": INPUT "RIGHT OR LEF
T (F/B/R/L) : ";ANS$
3305 ANS$ = LEFT$(ANS$,1): IF A
NS$ < > "F" AND ANS$ < > "
B" AND ANS$ < > "R" AND ANS
$ < > "L" THEN PRINT "TYPE
IN 'F' OR 'B' OR 'R' OR 'L'
": GOTO 3300
3310 IF ANS$ = "F" THEN X = 0:Y =
- 1: GOTO 3320
3311 IF ANS$ = "B" THEN X = 0:Y =
1: GOTO 3320
3312 IF ANS$ = "R" THEN X = 1:Y =
0: GOTO 3320
3313 IF ANS$ = "L" THEN X = - 1
:Y = 0: GOTO 3320

```



```

3320 SX = XP:SY = YP
3325 SX = SX + X:SY = SY + Y: PRINT
    "ZAP--";
3330 IF SX < 1 OR SX > 15 OR SY <
    1 OR SY > 15 THEN PRINT "FI
    ZZLE...": PRINT "THE SHOT LE
    FT THE MAZE.": PRINT "THE SH
    OT MISSED THE TWINKY!": GOTO
    3500
3335 IF MA(SX,SY) = 0 OR MA(SX,S
    Y) = 2 OR MA(SX,SY) = 3 OR M
    A(SX,SY) = 4 THEN 3325
3345 IF MA(SX,SY) = 1 THEN PRINT
    "BLAST !!": PRINT "THE SHOT
    HIT A WALL": PRINT "THE SHOT
    MISSED": GOTO 3500
3350 PRINT "OUCH !!!": PRINT "TH
    E SHOT HIT THE TWINKY": PRINT
    "THE TWINKY RETREATS"
3355 MA(SX,SY) = ST:XT = FN R(15
    ):YT = FN R(15):ST = MA(XT,
    YT):MA(XT,YT) = 5: GOTO 3500

3500 REM *** TWINKY MOVE LOGIC
3520 DT = FN A( SQR ((XT - XP) ^
    2 + (YT - YP) ^ 2))
3521 DO = FN A( SQR ((XP - XO) ^
    2 + (YP - YO) ^ 2))
3525 PRINT : PRINT "THE TWINKY I
    S "DT" UNITS AWAY."
3526 PRINT "THE OBJECTIVE IS "DO
    " UNITS AWAY."
3527 PRINT : PRINT "THE TWINKY M
    OVES ...": FOR I = 1 TO 500:
    NEXT I
3528 IF DT < 2 THEN PRINT : PRINT
    "<<<< S C H L O O R P ! ! !
    >>>>": PRINT "YOU'VE BEEN
    ABSORBED BY THE TWINKY !!!":
    PRINT "YOU LOSE.":WL = 1: GOSUB
    3600: RETURN
3530 IF XP < XT THEN X = - 1:Y =
    0: GOTO 3540

```

```

3531  IF XP > XT THEN X = 1:Y = 0
      : GOTO 3540
3532  IF YP < YT THEN X = 0:Y = -
      1: GOTO 3540
3533  IF YP > YT THEN X = 0:Y = 1
      : GOTO 3540
3540  MA(XT,YT) = ST:XT = XT + X:Y
      T = YT + Y:ST = MA(XT,YT):MA
      (XT,YT) = 5: GOTO 3020
3600  FOR I = 1 TO 40:XX = PEEK
      ( - 16336) + PEEK ( - 16336
      ) + PEEK ( - 16336): FOR J =
      1 TO 5: NEXT J,I: RETURN
3990  RETURN

```

# GAMES APPLES PLAY

**GAMES APPLES PLAY ON DISK!**  
**!!! Save hours of typing and de-bugging !!!**  
**ONLY \$15.95 WITH THIS COUPON**  
**and...**  
**ONLY available to readers of this book**

- \* Spend your time playing the games and writing your own versions
- \* Use the sound routines and graphics in your own programs
- \* Games may be copied, listed and modified
- \* Let your fingers do the playing, not the repetitious typing!



Please RUSH me the GAMES APPLES PLAY diskette for \$15.95

Please include \$1.75 for shipping and handling

California residents add 6.5% tax (\$1.04)

NAME \_\_\_\_\_

ADDRESS \_\_\_\_\_

CITY \_\_\_\_\_ STATE \_\_\_\_\_ ZIP \_\_\_\_\_

PHONE (     ) \_\_\_\_\_

☐ MASTERCARD

☐ VISA

☐ CHECK

CREDIT CARD NUMBER \_\_\_\_\_ EXPIRES \_\_\_\_\_

SIGNATURE (for charge cards) \_\_\_\_\_

Send to:  **DATAMOST™**  
8943 Fullbright Avenue  
Chatsworth, CA 91311  
(213) 709-1202





# WHAT BETTER WAY TO LEARN PROGRAMMING THAN BY PLAYING GAMES?

The authors have compiled a selection of classic Apple games along with many clever new games. Written in Applesoft BASIC, they are formatted in a way which lets you adapt any of the routines to your own programs. Also, each game is explained in easy-to-understand terms allowing you to modify and customize to your heart's content!

You'll learn principles of text formatting, word games, data statements, and input routines. Many of the games will help you understand how grids are constructed and graphics animated. Techniques are given which allow you to dissect Basic programs and see what makes them tick.

You'll have hours of fun learning to program in this easy, enjoyable way. And when you're done, you'll have the games to play! Whoever thought that gaming could teach so much about programming and de-bugging?



**RESTON PUBLISHING COMPANY, INC.**  
*A Prentice-Hall Company*  
Reston, Virginia

ISBN 0-8359-2417-3

## GAMES APPLES PLAY

From



8943 Fullbright Avenue  
Chatsworth, CA 91311  
(213) 709-1202



Reston  
Publishing  
Company  
Inc.